



Universidad
Carlos III de Madrid

Departamento de Informática

Ingeniería Informática

PROYECTO FIN DE CARRERA

TEACHAPP

Autor: José Manuel Maldonado Muñoz

Tutor: Javier García Guzmán

Título: TeachApp

Autor: José Manuel Maldonado Muñoz

Director: Javier García Guzmán

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 25 de Octubre de 2013 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

AGRADECIMIENTOS

A mi tutor por guiarme siempre con mucha mano izquierda en la visualización y consecución de los objetivos del proyecto.

A Paloma Leufuman, por apoyarme, empujarme y motivarme cuando todo era más oscuro y no se veía el final. Gracias por estar siempre ahí.

Y a mi familia por animarme a estudiar una carrera que me diera un futuro y que me sigue ilusionando como el primer día.

ÍNDICE DE CONTENIDOS

1.	Introducción	10
1.1.	Términos y definiciones	10
1.2.	Abreviaturas y Acrónimos	13
1.3.	Problemática	14
1.4.	Motivación.....	15
1.5.	Objetivos	16
1.6.	Metodología de Trabajo.....	17
1.6.1.	Metodología aplicada.....	17
1.6.2.	Identificación del cliente	18
1.6.3.	Roles	19
1.6.4.	Actividades	19
1.7.	Visión General del Documento	20
1.7.1.	Estado del Arte	20
1.7.2.	Requisitos	20
1.7.3.	Diseño.....	20
1.7.4.	Pruebas.....	21
1.7.5.	Conclusiones.....	21
2.	Estado del Arte	22
2.1.	¿Qué es E-learning?.....	22
2.2.	Historia	23
2.3.	Proyectos E-learning.....	24
2.3.1.	Aula Global	25
2.3.2.	IDoceo.....	26
2.3.3.	Duolingo	27
2.3.4.	Hangout	28
2.4.	Puntos a mejorar	29
2.5.	Entorno y tecnologías de Desarrollo	29
3.	Requisitos	31

3.1.	Descripción General	31
3.1.1.	Perspectiva del producto.....	31
3.1.2.	Funciones del producto	32
3.1.3.	Características de los usuarios	33
3.1.4.	Restricciones.....	34
3.1.5.	Suposiciones y dependencias.....	34
3.1.6.	Requisitos futuros.....	35
3.2.	Requisitos Específicos.....	35
3.2.1.	Interfaces Externas.....	36
3.2.2.	Funciones.....	37
3.2.3.	Restricciones de Diseño.....	40
4.	Diseño.....	42
4.1.	Arquitectura	42
4.1.1.	Justificación De La Descomposición Arquitectónica	42
4.2.	Modelo de Datos.....	45
4.2.1.	Diagrama de Entidad Relación E/R.....	46
4.2.2.	Modelo Relacional.....	47
4.3.	Vista de Implementación	48
4.3.1.	Vista	48
4.3.2.	Controlador	50
4.3.3.	Modelo	52
4.3.4.	Infraestructura.....	53
4.4.	Vista de Ejecución	54
4.5.	Vista de Despliegue	56
4.6.	Alternativas de diseño.....	57
4.7.	Diseño detallado.....	58
4.7.1.	Login	59
4.7.2.	Vista Lección	63
4.7.3.	Matriculación De Cursos	67
4.7.4.	Calificaciones	70
4.7.5.	Vista Cronograma	74
4.7.6.	Nuevo Cronograma	77

4.7.7. Editar Cronograma	80
5. Pruebas.....	84
5.1. Definition of Done	84
5.2. Procedimiento de validación de una tarea	85
6. Conclusiones.....	91
6.1. Dificultades.....	91
6.2. Consecución de objetivos.....	92
6.3. Aportaciones al E-Learning.....	93
6.4. Evolución futura	94
6.5. Conclusiones generales	95
7. Referencias	96
7.1. E-Learning.....	96
7.2. Tecnologías.....	97
7.3. Metodologías	98
7.4. Estado del arte	98
8. ANEXO I: Checklist funcional	99
9. ANEXO II: Manual del usuario	102
9.1. Login	102
9.2. Logout.....	103
9.3. Matriculación de Cursos.....	104
9.4. Vista de Lección.....	105
9.5. Evaluaciones	106
9.6. Ver cronograma.....	108
9.7. Nuevo Cronograma	109
9.8. Editar el Cronograma	110
9.9. Resumen.....	111
10. ANEXO III: Presupuesto	112
10.1. ¿Qué se presupuesta?	112
10.2. Parámetros generales	112
10.3. Costes de Personal	113
10.4. Amortización de equipos.....	114
10.5. Subcontratación de tareas	114

10.6. Costes de Funcionamiento.....	115
10.7. Resumen de costes.....	116

ÍNDICE DE FIGURAS

Ilustración 1: E-Learning	23
Ilustración 2: Euler	24
Ilustración 3: Aula Global.....	25
Ilustración 4: Idoceo	26
Ilustración 5: Duolingo	27
Ilustración 6: Hangouts.....	28
Ilustración 7: Ámbito del proyecto	33
Ilustración 8: Modelo de componentes.....	44
Ilustración 9: Modelo E/R.....	46
Ilustración 10: Modelo Relacional	47
Ilustración 11: Struts.....	49
Ilustración 12: Diagrama de Flujo	54
Ilustración 13: Vista de Despliegue.....	56
Ilustración 14: Diagrama de Pantallas	58
Ilustración 15: Login Facebook.....	59
Ilustración 16: Login	59
Ilustración 17: UML login.....	60
Ilustración 18: Diagrama de secuencia Login	62
Ilustración 19: Vista lección	63
Ilustración 20: UML Vista Lección	65
Ilustración 21: Diagrama de secuencia guardar Fichero	66
Ilustración 22: Vista Matriculación de cursos.....	67
Ilustración 23: UML matriculación de cursos.....	68
Ilustración 24: Diagrama de secuencias de matriculación de curso	69
Ilustración 25: Vista de calificaciones Profesor	70
Ilustración 26: Vista de calificaciones Alumno.....	70
Ilustración 27: UML Calificaciones	71
Ilustración 28: Diagrama de secuencias ver calificaciones.....	73
Ilustración 29: Vista cronograma.....	74
Ilustración 30: UML vista cronograma	75
Ilustración 31: Diagrama de secuencia detalle cronograma	76
Ilustración 32: Vista nuevo cronograma	77
Ilustración 33: UML nuevo cronograma.....	78
Ilustración 34: Diagrama de secuencia crear curso.....	79
Ilustración 35: Vista editar Cronograma superior	80
Ilustración 36: Vista editar Cronograma inferior	80
Ilustración 37: UML edición cronograma.....	82
Ilustración 38: Diagrama de secuencia de exportación a ICS	83

1. Introducción

1.1. Términos y definiciones

Término	Definición
Backlog	Cola de Historias de Usuario pendientes de planificar en Sprints, priorizadas por el Product Owner.
Checklist	Lista de elementos a verificar o comprobar.
Coach	Persona encargada de dirigir, instruir y entrenar a una persona con el objetivo de conseguir alguna meta o de desarrollar habilidades específicas.
css	Lenguaje que permite aplicar estilos y formato a un documento html
Daily	Breve reunión diaria realizada en metodologías ágiles como Scrum, en la que el equipo se pone al día en el avance de las tareas planificadas.
Demo	Versión de evaluación para mostrar y probar las funcionalidades del sistema.
Desarrollo en Cascada	Modelo de creación de software basado en etapas que deben de ser abordadas y resueltas cronológicamente.
Desarrollo Iterativo	Modelo de creación de software que permite la retroalimentación y revisión de las diferentes etapas del desarrollo.
DoD	Definition of Done (Definición de hecho). Reglas que marcan los requisitos que debe de cumplir una tarea para ser considerada como realizada.
Dropbox	Herramienta online en el ámbito de la nube, dedicada a almacenar y compartir archivos.
E-learning	Electronic-Learning (Aprendizaje electrónico). Concepto referido al aprendizaje mediante una presencia diferida en espacio y a veces en tiempo apoyado por el uso de herramientas tecnológicas que lo hagan posible.
Facebook	Red Social ampliamente conocida y usada.
Facilitador	Persona encargada de velar por el correcto funcionamiento del proceso, ayudando a resolver y apartar obstáculos.
FrameWork	Marco de trabajo.
Hangout	Herramienta de Google para la videoconferencia interactiva que permite entre otras cosas compartir pantallas y trabajar colaborativamente así como la grabación y difusión de contenido.

Historias de Usuario	Tarea a desarrollar basada en la petición funcional y en lenguaje coloquial de un cliente.
Hibernate	Herramienta que facilita el mapeo y acceso de atributos entre una base de datos relacional y un modelo de objetos de una aplicación.
html	Lenguaje estándar para la estructuración de contenido web.
Kanban	Metodología Ágil que sigue los principios del just in time.
Just in time	Modelo de trabajo en el que se producen los elementos en la cadena bajo demanda.
Manifiesto Ágil	Resumen de postulados en los que se basan las metodologías ágiles.
Maven	Herramienta para la gestión y construcción de proyectos Java que entre otras funcionalidades se encarga de las dependencias del proyecto.
Metodología	Conjunto de procedimientos racionales utilizados para alcanzar una gama de objetivos que rigen una exposición científica.
Metodología Ágil	Técnica de desarrollo software basada en el desarrollo iterativo e incremental.
Metodología Tradicional	Técnicas clásicas basadas en el análisis y definición predictiva de requisitos para diseñar y construir un modelo detallado de lo que se va a desarrollar. Suelen seguir un modelo de desarrollo en Cascada.
Metrica 3	Metodología de planificación, desarrollo y mantenimiento promovida por el ministerio de administraciones públicas.
MySQL	Sistema de gestión de bases de datos relacionales.
MVC	Model-View-Controller (Modelo-Vista-Controlador). Modelo de diseño software que independiza las 3 capas de la aplicación.
Plan Bolonia	Proceso de convergencia a nivel Europeo de los estudios universitarios con el objetivo de renovar y unificar los criterios educativos de los distintos países.
PlanBox	Herramientas de gestión de historias de usuario en el marco del desempeño del trabajo de una metodología Ágil.
Plugin	Es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande
Product Owner	Rol definido en Scrum encargado de la priorización y gestión de las historias de usuario.
Retrospectiva	Reunión realizada al final del Sprint para analizar los problemas acaecidos y plantear compromisos para la nueva iteración.
Scrum	Metodología Ágil basada en equipos multidisciplinares y autogestionados.

Scrum Master	Rol definido en Scrum encargado de velar por el cumplimiento de la metodología y por ejercer de facilitador del equipo.
Spring	Es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java
Sprint	Periodo de tiempo de desarrollo en metodologías ágiles que comienza con una planificación de tareas y termina con una retrospectiva.
Struts	Herramienta tecnológica para la creación de plataformas web que ayuda al programador en la tarea de definir un modelo MVC.
Tomcat	Servidor de aplicaciones con soporte de Servlets y JSPs.
Twitter	Popular Red Social para publicar textos de no más de 140 caracteres.
UML	Modelo de representación gráfica de elemento de código de programación.
XP	eXtreme Programming (Programación Extrema). Metodología ágil que pone énfasis en la adaptabilidad frente a la previsibilidad.
WEB 2.0	Concepto que define la interacción entre usuarios en una plataforma Web. Su máximo exponente son las redes sociales.

1.2. Abreviaturas y Acrónimos

Término	Significado
API	Application Programming Interface (Interfaz de Programación de Aplicaciones)
bbdd	Base de datos
css	Cascading Style Sheets (hojas de estilo en cascada)
DAO	Data Access Object (Objeto de Acceso a Datos)
DoD	Definition of Done (Definición de Hecho)
E-Learning	Electronic-Learning (Aprendizaje Electrónico)
E/S	Entrada y Salida
html	HyperText Markup Language (lenguaje de marcado hipertextual)
IOS	iPhone Operative System (Sistema Operativo de iPhone)
jsp	JavaServer Page (Página de servidor Java)
MVC	Model-View-Controller (Modelo-Vista-Controlador)
PFC	Proyecto de Fin de Carrera
POM	Project Object Model (modelo de objetos del proyecto)
UML	Unified Modeling Language (Modelo unificado de lenguaje)
XML	eXtensible Markup Language (lenguaje de marcas extensible)
XP	eXtreme Programming (Programación Extrema)
WAR	Web Application Archive (Archivo de aplicación Web)

1.3. Problemática

El objetivo de este proyecto es facilitar la labor educativa del profesorado en su ejercicio docente, agilizando las clases y facilitando el acceso a los recursos e información de las asignaturas impartidas. Por este motivo y debido al ámbito online de la aplicación por su orientación a internet en forma de plataforma Web, es claro que nos posicionamos en el sector del E-Learning.

El aprendizaje electrónico (E-Learning) se refiere a la educación virtualizada mediante el uso de las nuevas tecnologías y los recursos electrónicos propios de la sociedad de la información en la que vivimos. La educación trasciende la tradicional relación presencial entre profesor y alumno, permitiendo un aprendizaje a distancia mediante una selección y publicación de contenidos, apoyados por múltiples herramientas de muy diversos ámbitos, concebidas con el objetivo de facilitar la comunicación, evaluación y asimilación de conocimientos. Se trata de un sector en alza, muy favorecido por las bondades de la interacción de la Web 2.0, y aún por explotar en muchos aspectos, debido al auge de la demanda de este tipo de servicios.

El propósito del proyecto es explotar el potencial de la tecnología derivada de este sector, pero con el objetivo de solucionar algunas carencias y problemáticas presentes en la educación tradicional, siempre bajo el prisma de una orientación priorizada a la figura del profesor, sobre la recurrente fijación en las necesidades del alumno. Esto puede ser así puesto que la aplicación ha sido concebida como una herramienta para apoyar y agilizar una ponencia presencial en lugar de prescindir de ella y tratar de sustituirla por una simple mentorización a distancia. Estos son los principales puntos que se pretenden abordar:

- ❖ Administración y gestión de los alumnos y de la burocracia propia de la actividad docente, pero que desvían el foco de la verdadera labor profesoral de la enseñanza. Se gestionará la matriculación, los calendarios, las lecciones, los recursos, las calificaciones.
- ❖ Aprovechamiento de las bondades de las herramientas del E-Learning, aplicadas a las lecciones presenciales con alumnos, con el objetivo de agilizar las clases y posibilitar nuevos métodos de evaluación y asimilación de conocimientos. Se integran redes sociales como Facebook y herramientas como DropBox. Además se soportará el uso de herramientas colaborativas como Google Drive y se incentivará el uso de tecnologías de difusión como Hangouts y Youtube.
- ❖ Captación y suscripción de alumnos, sin depender de una entidad física que exponga y publicite el curso. Al ser una aplicación web, puede ser buscada y encontrada por cualquier persona en cualquier lugar del mundo.

1.4. Motivación

Los profesionales de la educación han visto en las últimas décadas una evolución muy importante en el sector, que ha replanteado las formas y los métodos clásicos del profesorado. No olvidemos que la actividad docente lleva desarrollándose desde la antigüedad, mucho antes de que Sócrates y Platón renegaran de los sofistas o de que la Iglesia filtrara el conocimiento bajo la doctrina cristiana en la Edad Media. También acaeció la renovación del renacimiento con el nuevo concepto del Humanismo, la disciplina Barroca y el posterior colofón ilustrado del siglo XVIII. En la actualidad, el nacimiento de internet y las posibilidades que ofrecen las nuevas tecnologías, han propiciado, junto a otros muchos factores, la aparición de una poderosa corriente educativa que promueve la participación y el trabajo práctico. Como ejemplos de esta tendencia podemos citar el plan Bolonia en las universidades, o la promoción de la Formación Profesional por parte del Estado.

En este contexto de evolución en el sector, los profesores han tenido que realizar un importante esfuerzo de adaptación y de renovación de sus métodos:

- ❖ Han asumido nuevas responsabilidades que suponen un mayor esfuerzo burocrático y administrativo, como controlar ausencias, justificar actuaciones, publicar recursos, etc.
- ❖ Se han formado en el uso de nuevas tecnologías.
- ❖ Han modificado la manera de impartir la clases para hacerlas más dinámicas y participativas.
- ❖ Han tenido que adaptar sus bibliografías y recursos para que puedan ser accesibles a los alumnos de hoy, mediante el uso de diapositivas, referencias electrónicas o ejercicios resueltos.
- ❖ Se ha debilitado la figura de autoridad del profesor y el respeto de la sociedad en general sobre él, sobre todo en la educación obligatoria de colegios e institutos, lo que redundo en una mayor preocupación por mantener el orden y la disciplina de la clase. Esto ha supuesto un mayor peso en la gestión de los castigos y sanciones.

Con este panorama, se antoja imprescindible una herramienta orientada al profesor que le ayude a gestionar y administrar todas estas situaciones y responsabilidades presentes en su ejercicio docente durante las clases, permitiéndole centrar su esfuerzo en transmitir los conocimientos a los alumnos en lugar de dispersarse con responsabilidades secundarias.

1.5. Objetivos

Tras analizar las necesidades y las carencias de los profesores en el contexto de una renovación en las formas de ejercer la docencia, motivada por una mayor relevancia de la participación y el trabajo práctico de los alumnos; además de por el auge de nuevas tecnologías que posibilitan la extensión de la educación a nuevas dimensiones con las plataformas E-Learning, los principales objetivos del proyecto son:

1. Desarrollo de una plataforma Web orientada al profesor que facilite el ejercicio de otras responsabilidades durante las clases más allá de la mera transmisión de conocimientos a sus alumnos, que debe de ser el foco de atención de sus esfuerzos y energías. Esto incluye:
 - ◆ Gestión de alumnos.
 - ◆ Gestión de recursos y contenidos.
 - ◆ Gestión de lecciones y calendarios.
2. Potenciación de la actividad docente mediante el apoyo de herramientas tecnológicas para la difusión, asimilación y evaluación del conocimiento. Esto incluye:
 - ◆ Publicación y difusión de información y recursos tecnológicos del curso (Dropbox, Google Drive).
 - ◆ Interactuación con el alumno.
 - ◆ Integración con redes sociales.
 - ◆ Integración de tecnologías de difusión como Youtube o Hangouts.
3. Independencia de una plataforma tecnológica, a veces inexistente, sujeta a la entidad de la que depende el curso. Esto permite:
 - ◆ Captación de alumnos interesados en la materia que se oferta dentro y fuera de los círculos de la entidad que da soporte al curso.
 - ◆ Universalidad de soportes físicos para el acceso a la información, puesto que internet puede ser accedido por tablets, smartphones y ordenadores convencionales, independientemente del sistema operativo que empleen.
 - ◆ Publicidad y prestigio social en la red.

1.6. Metodología de Trabajo

Para llevar a buen término un proyecto en informática de relativa envergadura, tal y como se ha demostrado a lo largo de la historia, es necesario seguir una Metodología o pautas marcadas con el propósito de alcanzar unos objetivos con un rigor pseudocientífico y contrastado. Actualmente, existen variedad de técnicas para estructurar el desarrollo de software, pero hay dos corrientes usualmente reconocidas.

- ❖ **Metodologías Tradicionales:** técnica clásica basada en el análisis y definición de requisitos para diseñar y construir un modelo detallado de lo que se va a desarrollar, tratando de anticipar y subsanar posibles problemas que puedan aparecer en las sucesivas fases. Suele definir un modelo de Desarrollo en Cascada y se apoya en técnicas y herramientas como UML para la definición del proyecto. Un ejemplo de este tipo de metodologías es Metrica 3, promovida por la administración pública.
- ❖ **Metodologías Ágiles:** corriente más moderna de desarrollo software nacida debido al problema de la incapacidad para predecir el alcance de un proyecto en las fases tempranas del mismo. Se basa en un modelo de Desarrollo Iterativo en el que desarrolladores y clientes trabajan retroalimentándose en un flujo de trabajo que permita la visualización y rectificación del producto en fases tempranas del proyecto con el objetivo de alinear posiciones y construir el producto útil demandado. Algunos ejemplos de metodologías ágiles son: Kanban, Scrum o XP.

1.6.1. Metodología aplicada

Para elegir una metodología adecuada, debemos identificar y definir las peculiaridades y necesidades del proyecto, eligiendo la que mayor beneficio pueda aportarnos solucionando un mayor número de problemas. A continuación se definen algunas particularidades del proyecto que influirán en la toma de la decisión adecuada:

- ❖ **Dedicación no exclusiva:** El desarrollador es activo laboralmente y debe compaginar su labor profesional con el desempeño del proyecto, esto limita las horas de dedicación semanales.
- ❖ **Fecha de entrega final no fija:** No existe una fecha de entrega cerrada sobre la que planificar el proyecto, simplemente un acuerdo tácito y aproximado para dar una visión general de la amplitud del PFC.
- ❖ **Reuniones planificadas:** Debido a que el desarrollador no tiene contacto permanente con el tutor por no asistir al centro de estudios diariamente, se deben concertar citas o tutorías periódicas para la puesta en común de ideas

y trabajos. También existe la posibilidad de una comunicación más activa vía Email.

❖ Línea de proyecto amplia: Puesto que la dimensión de la idea del proyecto es mucho más amplia de lo que la completitud de un PFC exige, se debe acotar el alcance del mismo, centrando el foco en las funcionalidades básicas, cuya prioridad puede variar en el tiempo, dependiendo de la evolución del trabajo. Esto permite y exige cierta flexibilidad a la hora de abordar tareas.

❖ Experiencia en metodologías ágiles: El desarrollador realiza su actividad profesional bajo el procedimiento de Scrum, por lo que tiene experiencia diaria en este tipo de metodologías.

Con estos puntos, se extrae la necesidad de utilizar una metodología que no sea rígida en tiempos y requisitos, que permita un trabajo a distancia y que en general sea flexible y modelable.

Con los puntos y necesidades definidas, se llega a la conclusión de que necesitamos seguir la corriente ágil de las metodologías puesto que aportan la flexibilidad requerida en el desempeño del trabajo y tenemos la madurez y la experiencia necesarias para usarlas provechosamente.

Se partirá del modelo de la metodología Scrum, adaptándolo a las particularidades y necesidades del proyecto tal y como promueve el Manifiesto Ágil. Debido a varias limitaciones que sufrimos y que exponemos a continuación, Scrum no puede aplicarse directamente al desarrollo de un PFC. A continuación, se describen las singularidades de la metodología usada:

1.6.2. Identificación del cliente

La aplicación será desarrollada para el uso del personal docente, pero puesto que nos encontramos en el ámbito de un PFC, el destino final del proyecto es el Tribunal que evaluará el desempeño del desarrollador. Por tanto, identificaremos al Tribunal como el cliente de nuestro producto.

1.6.3. Roles

En Scrum existen 3 tipos de roles además de los clientes, nosotros definiremos 2 roles debido al número limitado de personas que forman parte del proyecto.

- ❖ Tutor: Desempeña labores de Scrum Master y de Product Owner. Su labor consiste en definir y priorizar las Historias de Usuario y ejercer de facilitador para ayudar al desarrollador en el desempeño de sus tareas. Tiene un fuerte componente de Coach en el equipo. Este papel lo desempeña el profesor que tutoriza el PFC.
- ❖ Alumno: Desempeña labores de Desarrollador y de Scrum Master. Definirá el alcance de las historias de usuario y administrará el desarrollo del producto. Este papel lo desempeña el alumno que realiza el PFC.

1.6.4. Actividades

Daily: No se lleva a cabo debido a la imposibilidad de reunirse diariamente para exponer el avance de las tareas. Además debido a que el alumno no tiene dedicación exclusiva, no se realizan tareas diarias. Se usará la herramienta PlanBox para gestionar y seguir el trabajo realizado durante el Sprint. Documentando el esfuerzo empleado y la evolución del trabajo.

Planificación de Sprint: Se planifica el Sprint de 3 semanas tras la finalización del anterior en una reunión presencial del Tutor y el Alumno. Se recogen las tareas priorizadas contenidas en el Backlog y se estima la cantidad del trabajo que se podrá abordar y el alcance del mismo.

Administración del Backlog: La realizan los dos roles, definiendo las historias de usuario que vayan apareciendo, pero son priorizadas por el Tutor. Se administrará mediante la herramienta PlanBox.

Retrospectiva: Se unifica a la Planificación del Sprint, comentando los problemas y avances del Sprint anterior y amoldando la metodología a las necesidades del momento.

1.7. Visión General del Documento

En esta sección se describirán los distintos apartados del documento y la información que podrá encontrarse en ellos.

1.7.1. Estado del Arte

En este apartado del documento justificaremos en qué sector se mueve el proyecto y profundizaremos en la historia y evolución del mismo. Asimismo, se citarán los avances más significativos que se hayan realizado hasta la fecha y se buscarán carencias y aspectos a mejorar en el mismo.

Se citarán otros proyectos que se mueven en un ámbito similar al nuestro y se describirán sus características y debilidades.

Por último, se explicará cómo nuestro proyecto pretende solucionar algunas de las carencias encontradas y qué ventajas competitivas tiene respecto a otros proyectos en esos aspectos.

1.7.2. Requisitos

En este apartado del documento se identificará la perspectiva del producto y su funcionalidad, definiendo los usuarios de la aplicación y sus características. También se plantearán las suposiciones necesarias para realizar el producto y se listarán las dependencias externas del mismo.

Una vez hecho lo anterior, se especificarán los requisitos de la aplicación a nivel de interfaz, funcionalidad y rendimiento. Especificando en este punto las restricciones que se aplicarán al sistema.

1.7.3. Diseño

En este apartado se justificará la arquitectura del sistema y se detallarán sus componentes. Se hará una breve introducción a las tecnologías utilizadas para su programación (Spring, Struts, Hibernate, Maven, etc.) y se justificará su utilización en el mismo.

Se especificará el modelo de la aplicación, poniendo especial atención a su persistencia en base de datos y aportando diagramas relacionales de la estructura de las tablas donde se

almacenará la información. También se dará una perspectiva física indicando el hardware que alojara los distintos componentes cliente-servidor del sistema.

Por último, se hará un repaso página a página de las funcionalidades de la aplicación, mostrando una captura de pantalla del navegador y explicando sobre un modelo UML las posibilidades y estructura del código. También se incluirá un diagrama de secuencia en el que se especificarán las acciones que se realizan en el servidor al realizar una de las funcionalidades ofrecidas por la página detallada.

1.7.4. Pruebas

En éste área del documento se especificarán las pruebas de validación y aceptación que la aplicación deberá superar para poder ser entregada en cada una de sus fases o Sprints. Se debe comprobar que la adicción o mejora de funcionalidades no afecte al correcto funcionamiento y rendimiento de las ya implementadas, por lo que se expondrá una lista a modo de Checklist con las funcionalidades soportadas por la aplicación.

Se definirán asimismo unas Demos o visualizaciones por parte del profesor para la validación de la aplicación en cada reunión con él, de forma que puedan detectarse fallos en fases tempranas del desarrollo.

1.7.5. Conclusiones

En las conclusiones se exponen las dificultades acaecidas durante el proyecto y el grado de consecución de objetivos del mismo. Se recopilan las experiencias derivadas del método de trabajo empleado y del uso de las tecnologías definidas. Se hace balance del trabajo de investigación realizado y de las aportaciones del proyecto al sector del E-Learning; asimismo, se exponen los puntos de evolución futura en el área y las carencias de los actuales servicios. Por último se plantea una opinión personal al respecto en la que se enumeran las aportaciones del proyecto a la evolución académica y personal del alumno.

2. Estado del Arte

El E-Learning es un área en plena renovación tecnológica que está ejerciendo una gran influencia en la Educación tradicional, forzando a esta última a una adaptación de sus formas y métodos para posibilitar una sucesiva integración de las bondades que el aprendizaje electrónico nos aporta. Existen multitud de plataformas online, dependientes o no de instituciones educativas, que ofrecen una gran variedad de servicios y contenidos que han ido adaptándose desde el formato tradicional al electrónico. Esto se ha ido consiguiendo tras un largo proceso que ha requerido de la implicación y el esfuerzo de la comunidad educativa, que ha visto en esta nueva forma de enseñar una oportunidad para transmitir mejor los conocimientos y llegar a las nuevas generaciones. También ha sido necesaria una importante evolución tecnológica que permitiera pasar del simple intercambio postal de lecciones practicado desde el siglo XVIII, a la interacción y mentorización personalizada que se busca en las modernas plataformas online. En los últimos años, con la proliferación de los Smartphones y Tablets, se está invirtiendo en la creación de aplicaciones para estos dispositivos que permitan el aprendizaje en cualquier momento y lugar.

2.1. ¿Qué es E-learning?

La definición del aprendizaje electrónico es controvertida y difícil de concretar, principalmente porque es un término en constante renovación que ha sufrido importantes metamorfosis de un tiempo a esta parte. Por ejemplo, si entendemos E-Learning como aprendizaje a distancia, un simple intercambio de cartas entre un mentor y un alumno podría considerarse como tal, pero sin embargo este uso del concepto se reserva para algo mucho más tecnológico. Podría decirse que el E-Learning se refiere a todo aquel aprendizaje realizado por medios tecnológicos, pero alguien podría argumentar que cualquier persona que leyera un artículo en Internet estaría sujeto a esta técnica. Por otra parte, en esta situación echamos en falta un mentor o un plan de estudios que guíe el aprendizaje, aunque, ¿realmente son estos elementos indispensables en la educación electrónica? Como vemos, es difícil ponerse de acuerdo y delimitar qué es E-Learning, porque diversos autores dan definiciones muy distintas según qué características o contextos tengan en cuenta. Nosotros definiremos el E-Learning como:

Concepto referido al aprendizaje estructurado mediante una presencia diferida en espacio y a veces en tiempo apoyado por el uso de herramientas tecnológicas que lo hagan posible.



Ilustración 1: E-Learning

2.2. Historia

Los inicios de la educación a distancia pueden situarse a principios del siglo XVIII cuando se tiene constancia de un ofrecimiento de envío de material de autoinstrucción a los interesados a través del correo postal, con la posibilidad de contar con tutorías por este mismo medio. Uno de los ejemplos más famosos en aplicar este método fue Euler.

La industrialización del siglo XIX y la demanda de obra cualificada después de la Segunda Guerra Mundial favorecieron el surgimiento de ofertas de educación abierta, que coincide con la inclusión de la modalidad a distancia en instituciones universitarias. También hay un fuerte impulso comercial de recursos didácticos para cubrir una demanda educativa en zonas de difícil cobertura. Los resultados de experiencias logran que la modalidad perdure y se enlace con el desarrollo e incorporación de tecnologías vinculadas a la telefonía, la radio y la televisión. El hecho de que los programas a distancia comiencen a complementar el material impreso con audio o vídeo, obligó a la profesionalización de los equipos dando lugar a la siguiente generación marcada por la explosión tecnológica y en la que se integran las telecomunicaciones con otros medios educativos a través de la informática.

El ordenador se convierte en un sistema interactivo en sí mismo, al sintetizar e integrar en sus programas: texto, imágenes, sonidos y animación. La llegada de Internet y el correo electrónico dan nuevo significado y poder a la educación a distancia generando la posibilidad de enseñar y aprender a través de la Red. El surgimiento de plataformas especializadas que aunaban las distintas técnicas y tecnologías disponibles marcaron la definitiva ruptura del E-Learning con la mera educación a distancia; proporcionando un auténtico contexto específicamente desarrollado para la difusión y asimilación de materiales diseñados y orientados para tal fin. Un último paso de esta evolución es la inclusión del concepto red social, en la que los alumnos pueden interactuar y retroalimentarse para un aprendizaje mucho más efectivo y menos dependiente del mentor. La posibilidad de acceder a estas plataformas de aprendizaje desde cualquier lugar con la tecnología de los Smartphones da un giro de tuerca más al concepto de E-Learning.



Ilustración 2: Euler

2.3. Proyectos E-learning

Existen muchas plataformas E-Learning en el mercado orientadas a muy diversos temas. Por proximidad, vamos a analizar tres de ellas: Aula Global, por ser la plataforma usada en la universidad Carlos III de Madrid, que es la universidad para la que se desarrolla el

presente proyecto; iDoceo, por ser una aplicación móvil, de forma que profundizaremos en las plataformas orientadas a un dispositivo determinado; Duolingo, que está orientado exclusivamente al aprendizaje de idiomas, es decir, es una plataforma especializada en un área temática; y por último los Hangouts de Google, como herramienta de E-Learning más que como una plataforma en sí misma.

2.3.1. Aula Global

Aula Global es la plataforma utilizada por la Universidad Carlos III de Madrid para la publicación de contenidos y de información online. Está dirigida a dos tipos de perfiles: alumnos y profesores. En ambos casos se fundamenta en una lista de asignaturas donde publicar recursos docentes. Se apoya de una serie de herramientas como son un calendario de eventos, foros, mensajes y calificaciones. También es posible subir los trabajos de los alumnos directamente a la plataforma entre otras características.

En verano de 2013 se lanza la nueva versión en Moodle 2.4 que ofrece grandes novedades respecto a su predecesora como: una nueva navegación, un archivo personal, chat, wikis y nuevos recursos como el denominado “El Libro” que indexa material didáctico de forma ordenada con la posibilidad de añadir contenido multimedia.

El principal problema de la plataforma es que depende de un organismo concreto, en este caso una Universidad, por lo que está dirigida exclusivamente a los profesores y alumnos de la Carlos III. Tampoco integra otras herramientas tecnológicas disponibles en la web como son las redes sociales y su visualización en móviles es complicada.



Ilustración 3: Aula Global

2.3.2. IDoceo

IDoceo es una aplicación móvil para IOS diseñada para dar soporte al profesor en su ejercicio docente. Entre otras características incluye comunicaciones con los alumnos vía email, además de varias herramientas que facilitan la gestión de la clase, ya sea para listar la información de los alumnos, para evaluarlos o simplemente para elegir uno de ellos al azar o ubicarlos en un aula. También contiene un calendario de clases y otras herramientas para la administración y el apoyo docente en el aula.

Las limitaciones de este proyecto son la escasa universalidad, dado que es accesible únicamente para profesores que dispongan de un producto Apple compatible previo pago de su coste, dado que no es gratuito. Al no poder usarse en un ordenador convencional y carecer de interfaz para el alumno, se están perdiendo muchas posibilidades al obligar a realizar la comunicación a través de correo electrónico, obviando la iniciativa bidireccional.



Ilustración 4: IDoceo

2.3.3. Duolingo

Duolingo es una plataforma disponible tanto para navegador web como para aplicación en Android e IOS. Se trata de una red social de aprendizaje de idiomas en la cual existen herramientas específicamente diseñadas para asimilar y practicar los idiomas: leyendo, escribiendo, escuchando y hablando. Se retroalimenta de los usuarios para complementar y corregir las lecciones con foros de discusión, reporte de errores y alternativas, además de fomentar el espíritu competitivo de sus usuarios con niveles y puntuaciones. Estructura los cometidos según su dificultad y ánima a repasarlos para que no se olviden.

Se trata de una plataforma independiente, universal y gratuita que se financia con la traducción de artículos por parte de la propia comunidad. Integra tecnologías como Facebook y Google para el inicio de sesión y contiene contenidos adaptados específicamente para el aprendizaje electrónico.

El principal problema de la aplicación es su especialización en el aprendizaje de idiomas, lo que limita la posibilidad de aventurarse en otros campos del aprendizaje, dado que contiene herramientas demasiado específicas para este ámbito y es poco moldeable.



Ilustración 5: Duolingo

2.3.4. Hangout

Hangout es una herramienta de Google liberada en 2013, similar a una videoconferencia, pero añadiendo la posibilidad de interactuar a través de aplicaciones y otras herramientas colaborativas. Además del ya conocido Google Drive, que permite entre otras cosas trabajar sobre distintos documentos de oficina de manera simultánea, los Hangouts permiten compartir la pantalla con el resto de personas para seguir por ejemplo una presentación. Además publica un API sobre el cual distintos desarrolladores pueden crear pequeñas aplicaciones para utilizar sobre esta plataforma, lo que la hace una herramienta de E-Learning bastante potente en cuanto a interacción y trabajo colaborativo. Existe la posibilidad de emitir en directo las sesiones e incluso grabarlas para su publicación en Youtube.

La principal debilidad de Hangout es que no ofrece un curso con continuidad en el tiempo, sino que depende de la organización del mentor y los alumnos, no teniendo de forma predeterminada una estructura que seguir fuera de la interacción puntual o de la visualización del video a posteriori. Es por ello, que puede considerarse una herramienta más que una plataforma de E-Learning en sí misma.

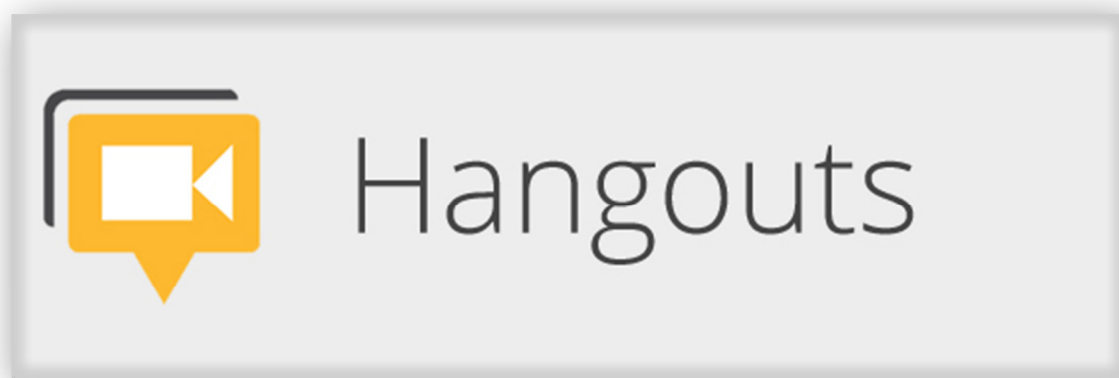


Ilustración 6: Hangouts

2.4. Puntos a mejorar

Como hemos visto tras el análisis de los distintos proyectos seleccionados, existen una serie de problemas que es necesario identificar y solventar. Los enumeramos:

- ❖ **Independencia:** Que una plataforma sea independiente de un centro de estudios permite que cualquier persona pueda acceder a ella y utilizarla libremente, abriendo el mercado de usuarios enormemente.
- ❖ **Universalidad:** Si una aplicación solo está disponible para una determinada arquitectura como es IOS, se limitará mucho su uso. Lo ideal es que pueda ser soportada por todos los dispositivos y arquitecturas mediante un navegador web.
- ❖ **Versatilidad:** Si una plataforma está demasiado especializada en un tipo de aprendizaje, difícilmente puede adaptarse a otro tipo de contenidos o necesidades. Lo ideal es que se utilicen herramientas lo suficientemente versátiles y potentes como para permitir la enseñanza de todo tipo de contenidos. Es necesaria la integración de redes sociales y otras herramientas tecnológicas que posibiliten esto.
- ❖ **Estructuración:** Es necesario ordenar el contenido y los recursos de los cursos para que pueda ser seguido de forma síncrona o asíncrona sin perder información. Es necesario conocer de antemano la duración y contenidos del curso, así como implementar un seguimiento del mismo, tanto cronológico como de evaluación.

2.5. Entorno y tecnologías de Desarrollo

Por los motivos expuestos en el anterior apartado, se hace necesaria una tecnología que provea independencia, universalidad y versatilidad. Además de permitir desarrollar una estructuración, es decir, una tarea de gestión de los cursos que sea eficaz y manejable. Actualmente, existen diversas alternativas en el mercado que pueden cumplir estos requisitos, las más importantes son:

.NET: Es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permite un rápido desarrollo de aplicaciones. Basado en el, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado. Este último punto es su principal desventaja, dado que lo hace muy corporativista.

J2EE: Es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. Actualmente es el entorno más utilizado y el que mayor universalidad y opciones ofrece, por lo que es nuestro preferido para el desarrollo.

Grails: Es un framework para aplicaciones web libre desarrollado sobre el lenguaje de programación Groovy (el cual a su vez se basa en Java). Grails pretende ser un marco de trabajo altamente productivo siguiendo paradigmas tales como convención sobre configuración o no te repitas (DRY), proporcionando un entorno de desarrollo estandarizado y ocultando gran parte de los detalles de configuración al programador. Todavía es poco usado y se le critica la falta de formalidad en su sintaxis, lo que hace que sea un lenguaje confuso y difícil de entender a ciertos niveles.

3. Requisitos

3.1. Descripción General

Se plantea la necesidad de una plataforma Web independiente del centro de estudios del que dependen los cursos, que pueda ser accedida por cualquier persona desde cualquier dispositivo, independientemente del sistema operativo que utilice. La aplicación se centrará en facilitar la labor docente del profesor, dando soporte administrativo y ayudando a la gestión de alumnos y contenidos. Se encargará de la publicación y difusión de la información relativa al curso y de los recursos del mismo. Facilitará la comunicación e interacción profesor-alumno y dará soporte a herramientas que apoyen la labor docente durante las clases.

Durante la realización del proyecto se seguirá una modificación de la metodología ágil: Scrum; que permita adaptar y flexibilizar el trabajo a la priorización de requisitos que vayan apareciendo en cada momento, realizando historias de usuario que puedan ser probadas y validadas en cada Sprint.

3.1.1. Perspectiva del producto

El producto está ubicado en el ámbito del E-learning, o aprendizaje electrónico. Su objetivo principal es facilitar la labor docente del profesor también durante las clases presenciales, por lo que no se limita a una plataforma tecnológica de intercambio de información, sino que aspira a convertirse en un instrumento administrativo y de gestión que ayudará al personal docente a focalizar sus esfuerzos en transmitir los conocimientos del curso en lugar de perder tiempo con obligaciones burocráticas secundarias.

Para ello, el sistema será publicado en internet en forma de plataforma web, y se relacionará con otras aplicaciones para ayudarse a cumplir alguna de las funcionalidades que se describirán. Así, se usará la red social Facebook para la identificación de los usuarios, apoyándonos en su tecnología para garantizar la confidencialidad y seguridad de su información personal. También usaremos los servicios de Dropbox para la publicación y difusión de recursos propios de los cursos, ahorrando así capacidad de conexión de nuestro servidor y facilitando la gestión documental.

3.1.2. Funciones del producto

El sistema deberá permitir identificarse mediante Facebook a los usuarios de forma que pueda realizarse un seguimiento de los mismos, ya sea desde la perspectiva del profesor, que publicará los cursos que imparte así como la información relevante del mismo y los recursos necesarios para el correcto seguimiento por parte de los alumnos. Asimismo, los alumnos tendrán una información académica que irá evolucionando según sea su progresión en la asignatura, esto incluye gestión de ausencias, de calificaciones, de tareas, etc.

Los profesores podrán crear cursos, para ello deberán ser capaces de crear un cronograma exportable en el que se indique la duración en semanas del mismo, así como los días y las horas en las que se impartirán las clases y el temario previsto para cada una de ellas. Se deberán tener en cuenta los días festivos y las vacaciones. También se añadirá información de la asignatura como: la descripción, la forma de evaluación o el horario para tutorías. Se podrán publicar los recursos necesarios para cada sesión docente, esto incluye cualquier tipo de archivo: imagen, pdf, documento office, etc.

Los alumnos podrán subscribirse a los cursos publicados, una vez hecho esto, podrán ver la información de la asignatura, así como acceder y descargar los recursos docentes que el profesor haya publicado. Los alumnos podrán ser calificados y gestionados por el profesor del curso: esto incluye control de asistencias y progresión de tareas.

El ámbito del proyecto deja fuera otras funciones importantes en una plataforma E-Learning, pero que por cuestiones de tiempo no se abordarán, tales como: la Mensajería, la potenciación de la plataforma como red social o diseñar una interfaz específica para cada dispositivo. A continuación, vemos un gráfico resumen de los módulos que quedan englobados o descartados de la primera implementación de TeachApp.

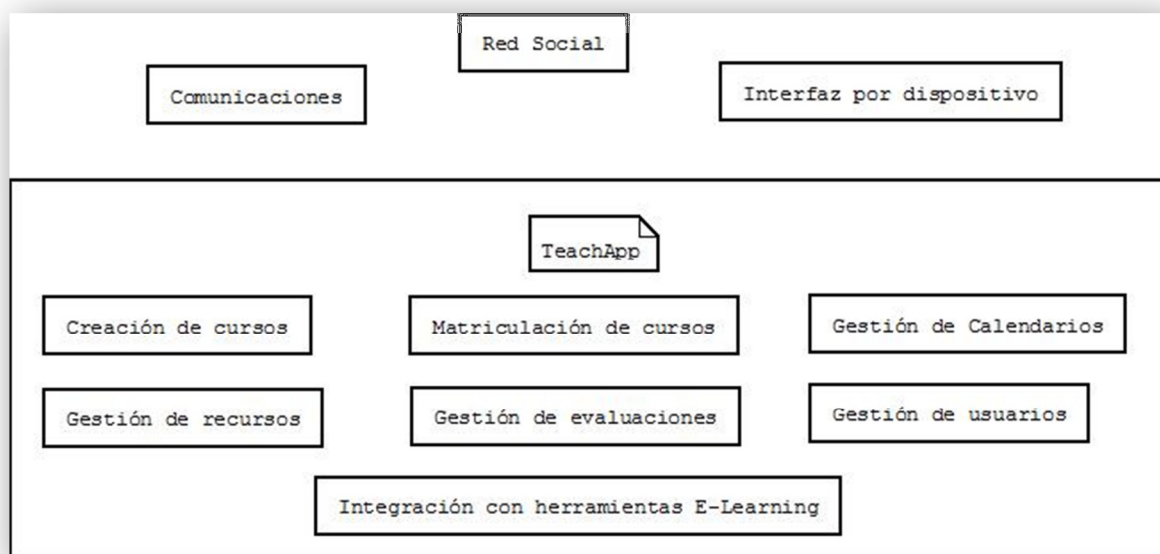




Ilustración 7: Ámbito del proyecto

3.1.3. Características de los usuarios

El sistema está dirigido a dos tipos de usuarios, siempre en un contexto de enseñanza y aprendizaje:

-  **Alumno:** Este usuario será una persona de edad variable pero predominantemente joven. No se le presupondrán conocimientos específicos en informática, pero sí cierta facilidad para moverse en entornos tecnológicos; como redes sociales. Dispondrá de un dispositivo con acceso a internet mediante navegador web. Su labor consistirá mayoritariamente en demandar y consumir información acerca de sus cursos.
-  **Profesor:** Este usuario será una persona de edad variable a partir de la veintena. Desempeña el papel de profesor dependiente de alguna institución educativa o por cuenta propia. Se le presuponen conocimientos y recursos de información amplios en uno o varios ámbitos y tiene la necesidad de transmitirlos a sus alumnos. No tiene conocimiento específicos en informática, pero sí cierta facilidad para moverse en entornos tecnológicos; como redes sociales. Dispondrá de un dispositivo con acceso a internet mediante navegador web. Su labor consistirá mayoritariamente en publicar información y recursos que puedan ser útiles para el desarrollo de los cursos. También se apoyará en la aplicación para gestionar la labor administrativa.

3.1.4. Restricciones

La aplicación deberá cumplir la normativa legal en cuanto a confidencialidad y seguridad de la información, por tanto, será necesario disponer de usuarios únicos que serán autenticados mediante Facebook, dado que no se desea implementar una gestión de usuarios avanzada en la aplicación sino aprovechar una ya existente y reconocida.

También se quiere externalizar el almacenaje y gestión de los recursos, por lo que se usará un sistema externo que ofrezca estos servicios en lugar de dedicar espacio de almacenaje y recursos de conexión en el servidor de la aplicación.

Los cronogramas generados por la aplicación deben ser exportables en un formato estándar a otras aplicaciones, de forma que puedan ser reusados fácilmente en los calendarios usados habitualmente por los usuarios.

Se usará tecnología reconocida en sus versiones más modernas con el objetivo de facilitar el mantenimiento y alargar la vida útil del producto. Esta tecnología deberá ser portable y universal, es decir, deberá permitir el funcionamiento de la aplicación en cualquier dispositivo con acceso a internet mediante un navegador moderno estándar, por lo que se usarán lenguajes de programación que no sean específicos para una arquitectura o sistema operativo.

3.1.5. Suposiciones y dependencias

Para la formulación de los requisitos del sistema, se han presupuesto ciertas condiciones que en el caso de cambiar podrían afectar al correcto funcionamiento de la aplicación. Dichas dependencias pasan a enumerarse a continuación:

- ❖ Existencia de Facebook como empresa dominante en el sector de las redes sociales con una importante presencia de usuarios activos en internet. Publicación por parte de esta empresa del API necesario para que un usuario pueda autenticarse de forma segura en su aplicación, de forma que nuestro sistema tenga garantías sobre la identidad del usuario.
- ❖ Existencia del servicio de 'nube' DropBox. Se deberá vigilar el cambio y depreciación de las funcionalidades ofrecidas en su API v1.0.
- ❖ Disposición de las plataformas Hardware necesarias para la instalación y publicación de la aplicación. Esto incluye servidor de aplicaciones web configurado para el acceso mediante internet de los usuarios del sistema y

todas la tecnologías necesarias para levantar y correr la aplicación, las cuales se definirán más adelante en los requisitos técnicos del sistema.

3.1.6. Requisitos futuros

El sistema podrá usarse como un módulo adicional de una plataforma dependiente de una entidad educativa ya existente, por lo que deberá emplearse una arquitectura que permita dicha modularidad. Además la aplicación debe diseñarse para alargar el ciclo de vida del producto más allá de la entrega inicial del mismo, mediante mantenimiento evolutivo e implementación de nuevas características y herramientas que escapen al ámbito actual del producto. Al emplear varios servicios externos, puede ser necesario migrar a una versión más moderna de los mismos que solventen funcionalidades o errores que puedan afectar al sistema, así como adaptar las interfaces de métodos deprecados.

3.2. Requisitos Específicos

Cada requisito especificado tendrá un formato similar al de la siguiente tabla, en ellos podemos encontrar la siguiente información:

Identificador	RU[I,F,R]-XX
Descripción	(descripción)
Usuario	[Profesor, Alumno]*
RUF afectados	[RU-XX-YY,RUXX,TODOS]*
Fuente	[Product Owner, Developer Team]

- **Identificador:** Podrá ser de tres tipos: RUI, para referirse a los requisitos de interfaz; RUF, para los requisitos funcionales; y RUR para los de restricción. Además se añadirá un número identificativo de dos cifras para identificarlo unívocamente.
- **Descripción:** define a qué se refiere el requisito.
- **Usuario:** indica el rol que podrá desempeñar la funcionalidad. Solo tiene sentido en los RUF, por lo que es una especificación opcional.

- **RUF afectados:** Indica los requisitos funcionales que se ven afectados por las restricciones. Solo tienen sentido en los RUR, por lo que es una especificación opcional.
- **Fuente:** Indica quién es el autor del requisito, es decir, quién planteó la necesidad

3.2.1. Interfaces Externas

Identificador	RUI-01
Descripción	Los usuarios deben interactuar con la aplicación a través de navegadores modernos: Internet Explorer 10, Google Chrome v.29 y Mozilla Firefox v.24
Fuente	Product Owner

Identificador	RUI-02
Descripción	Integrar API oficial de Facebook para conectar TeachApp con dicha red social y poder utilizar su servicio de login.
Fuente	Product Owner

Identificador	RUI-03
Descripción	Integrar API oficial de Dropbox para conectar TeachApp con dicho servicio de almacenamiento en nube de ficheros y poder utilizar sus servicios.
Fuente	Product Owner

Identificador	RUI-04
Descripción	Integrar Hibernate con TeachApp para delegar el manejo de la base de datos a sus servicios.
Fuente	Product Owner

3.2.2. Funciones

Identificador	RUF-01
Descripción	Crear cronograma del curso indicando: el nombre, la fecha de inicio y el número de semanas de duración.
Usuario	Profesor
Fuente	Product Owner

Identificador	RUF-02
Descripción	Modificar cronograma del curso indicando por lección: la descripción del temario previsto, los días de la semana y los horarios de las clases.
Usuario	Profesor
Fuente	Product Owner

Identificador	RUF-03
Descripción	Modificar cronograma del curso para marcar/desmarcar días festivos y vacacionales.
Usuario	Profesor
Fuente	Product Owner

Identificador	RUF-04
Descripción	Exportar el cronograma de un curso al formato estándar ICS para su visualización en calendarios.
Usuario	Profesor
Fuente	Product Owner

Identificador	RUF-05
Descripción	Añadir y Editar la información de los cursos impartidos por el usuario: descripción y evaluación.
Usuario	Profesor
Fuente	Product Owner

Identificador	RUF-06
Descripción	Publicar recursos y material docente de los cursos por lección utilizando Dropbox.
Usuario	Profesor
Fuente	Product Owner

Identificador	RUF-07
Descripción	Autenticar al usuario en la aplicación utilizando Facebook.
Usuario	Profesor y Alumno
Fuente	Product Owner

Identificador	RUF-08
Descripción	Permitir matricular los cursos entre la oferta de cronogramas existentes.
Usuario	Profesor y Alumno
Fuente	Product Owner

Identificador	RUF-09
Descripción	Visualizar la información de la próxima clase de entre los cursos seleccionados: día y hora de la clase, información del curso, temario previsto y recursos disponibles.
Usuario	Profesor y Alumno
Fuente	Product Owner

Identificador	RUF-10
Descripción	Visualizar resumen de las próximas lecciones de los cursos seleccionados y enlazar con la descripción completa de las mismas: fecha, hora, asignatura y temario.
Usuario	Profesor y Alumno
Fuente	Product Owner

Identificador	RUF-11
Descripción	Visualizar/descargar el material y los recursos docentes publicados en el curso para cada lección.
Usuario	Profesor y Alumno
Fuente	Product Owner

Identificador	RUF-12
Descripción	Gestionar las evaluaciones de los alumnos, registrando y editando los conceptos de evaluación, el porcentaje de peso en la asignatura de cada concepto, las calificaciones de cada concepto por alumno y los comentarios del profesor de forma pública (para el alumno) y privada (solo para el profesor).
Usuario	Profesor
Fuente	Product Owner

Identificador	RUF-13
Descripción	Visualizar las evaluaciones obtenidas en cada prueba realizada del curso.
Usuario	Profesor y Alumno
Fuente	Product Owner

3.2.3. Restricciones de Diseño

Identificador	RUR-01
Descripción	Limitar el acceso a las funcionalidades a los usuarios descritos para cada RUF, asegurando una sesión vigente y autenticada en la aplicación.
RUF afectados	Todos
Fuente	Product Owner

Identificador	RUR-02
Descripción	Disponer de un servidor web Tomcat donde alojar y publicar la plataforma.
RUF afectados	Todos
Fuente	Developer Team

Identificador	RUR-03
Descripción	Disponer de un servidor de base de datos MySQL donde almacenar los datos persistentes de la aplicación.
RUF afectados	Todos
Fuente	Developer Team

Identificador	RUR-04
Descripción	Utilizar Maven para la gestión de dependencias y el despliegue de la aplicación
RUF afectados	Todos
Fuente	Developer Team

Identificador	RUR-05
Descripción	Utilizar Struts como herramienta de soporte para el desarrollo de la interfaz de la aplicación
RUF afectados	Todos
Fuente	Developer Team

Identificador	RUR-06
Descripción	Utilizar Spring como framework para la administración y configuración de los elementos de la aplicación.
RUF afectados	Todos
Fuente	Developer Team

4. Diseño

Debido a la constante evolución de las plataformas E-Learning, es necesaria una arquitectura que permita una gran flexibilidad a la hora de añadir o modificar funciones. Para ello, la aplicación se apoya en probadas tecnologías Java que ayudan a conseguir dicho fin. Así, se parte de un modelo MVC soportado por la tecnología Struts, que permite la rápida creación y edición de jsp, actions y forms. Para el acceso a la información se usa un servidor de base de datos de MySql que es accedido con el soporte que proporciona la tecnología Hibernate. Por último para la administración de dependencias se usa Maven, configurado para desplegar la aplicación en un servidor Tomcat.

Se usan otros servicios con el objetivo de integrar funcionalidades en la aplicación. Así, por ejemplo, se tiene acceso al API de Facebook o de Dropbox para la respectiva gestión de usuarios y de recursos.

4.1. Arquitectura

4.1.1. Justificación De La Descomposición Arquitectónica

Antes de describir la arquitectura para el sistema, se hace necesaria una explicación de los criterios de selección que se han considerado:

El criterio más importante para seleccionar la arquitectura del sistema va a ser la modificabilidad. Es fundamental que el proyecto pueda adaptarse a otros escenarios, como otras plataformas de E-Learning, de tal forma que se pueda implantar de forma sencilla el sistema en dicho contexto.

También resulta muy deseable la extensibilidad del sistema. Añadir funcionalidades al sistema base se antoja imprescindible para ofrecer un mejor servicio respondiendo a las demandas que se vayan produciendo de forma que el sistema no quede obsoleto e inútil. Así por ejemplo, se prevé la inclusión de nuevas características y funcionalidades en el futuro.

El tercer criterio básico sería la eficiencia. Es vital que el sistema funcione de manera lo suficientemente rápida como para que sea útil para la función que lleva a cabo. Si se quiere

emplear como centro de información de los cursos, no tendría ningún sentido que los datos se demoraran más de la cuenta.

Existen más criterios que no se pueden ignorar, como el coste; sin embargo se asumirá que quedarán en un segundo plano, ya que al estar en el ámbito de un proyecto de fin de carrera, el objetivo del sistema es académico y su implantación en un sistema de producción real queda fuera del ámbito inicial del mismo.

Una vez tenidos en cuenta los criterios que más van a influir, se ha propuesto una arquitectura en cuatro capas que combine las ventajas del MVC y la arquitectura de capas “pura”. Dichas ventajas consideran la modificabilidad del sistema, ya que se independizará con respecto a la plataforma; y la extensibilidad, ya que se minimizan las dependencias. Esta última característica beneficia también al grado de acoplamiento entre los componentes, que va a ser reducido. No obstante, habrá una alta cohesión dentro de cada uno de ellos. Por otra parte, con esta arquitectura se facilita el mantenimiento del sistema, que es un punto muy a tener en cuenta en todo proyecto.

Como consecuencia negativa de esta decisión, se verá aumentada la complejidad en esta primera fase de implantación, ya que la arquitectura no resulta sencilla de implementar. No obstante, se cuentan con diversas tecnologías específicamente concebidas para facilitar esta labor arquitectónica como Hibernate, Spring y Maven, que pasaremos a explicar después.

El sistema quedará por tanto, descompuesto en cuatro capas o subsistemas que a continuación se comentan sucintamente:

- ❖ **Vista:** va a permitir a los usuarios comunicarse con la aplicación a través de interfaces de entrada y salida (Web).
- ❖ **Modelo:** va a contener las clases que aparecen en el modelo conceptual.
- ❖ **Controlador:** se encargará de intermediar entre la vista y el modelo. Traducirá las peticiones que hacen los usuarios, a través de las interfaces de entrada, a mensajes entendibles por el modelo y viceversa.
- ❖ **Infraestructura:** este subsistema proporciona interfaces para la comunicación con otros dispositivos o sistemas en los que se apoya la aplicación.

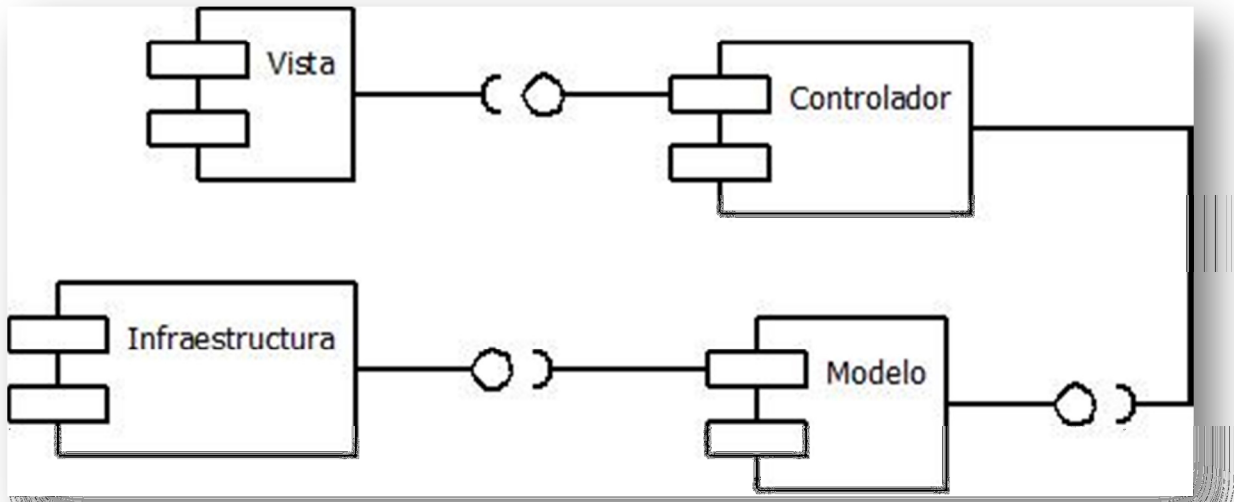


Ilustración 8: Modelo de componentes

Esta organización de la arquitectura genera una serie de dependencias que hay que manejar y organizar. Para esta labor, utilizaremos la tecnología Maven.

Maven

Maven es una herramienta de software para la gestión y construcción de proyectos Java. Es similar en funcionalidad a Apache Ant, pero tiene un modelo de configuración de construcción más simple, basado en un formato XML.

Maven utiliza un Project Object Model (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado.

Una característica clave de Maven es que está listo para usar en red. El motor incluido en su núcleo puede dinámicamente descargar plugins de un repositorio, el mismo repositorio que provee acceso a muchas versiones de diferentes proyectos de código abierto en Java, de Apache y otras organizaciones y desarrolladores. Maven está construido usando una arquitectura basada en plugins que permite que utilice cualquier aplicación controlable a través de la entrada estándar.

4.2. Modelo de Datos

Al trabajar con una aplicación suele ser habitual que surja la necesidad de almacenar datos para que persistan en el tiempo. Esto es más fácilmente reconocible cuando se trabaja con un lenguaje orientado a objetos como Java, donde cada concepto de negocio o entidad se verá reflejado como una clase en el código. Con el objetivo de identificar y conocer los objetos de negocio del ámbito del sistema, es necesario un estudio y una representación que permita un acceso efectivo y consistente a la información, ya sea en tiempo de proceso o para recuperar información almacenada.

A continuación se detallará el modelo de datos que se ha utilizado en la aplicación y su modelo relacional para su persistencia en BBDD.

4.2.1. Diagrama de Entidad Relación E/R

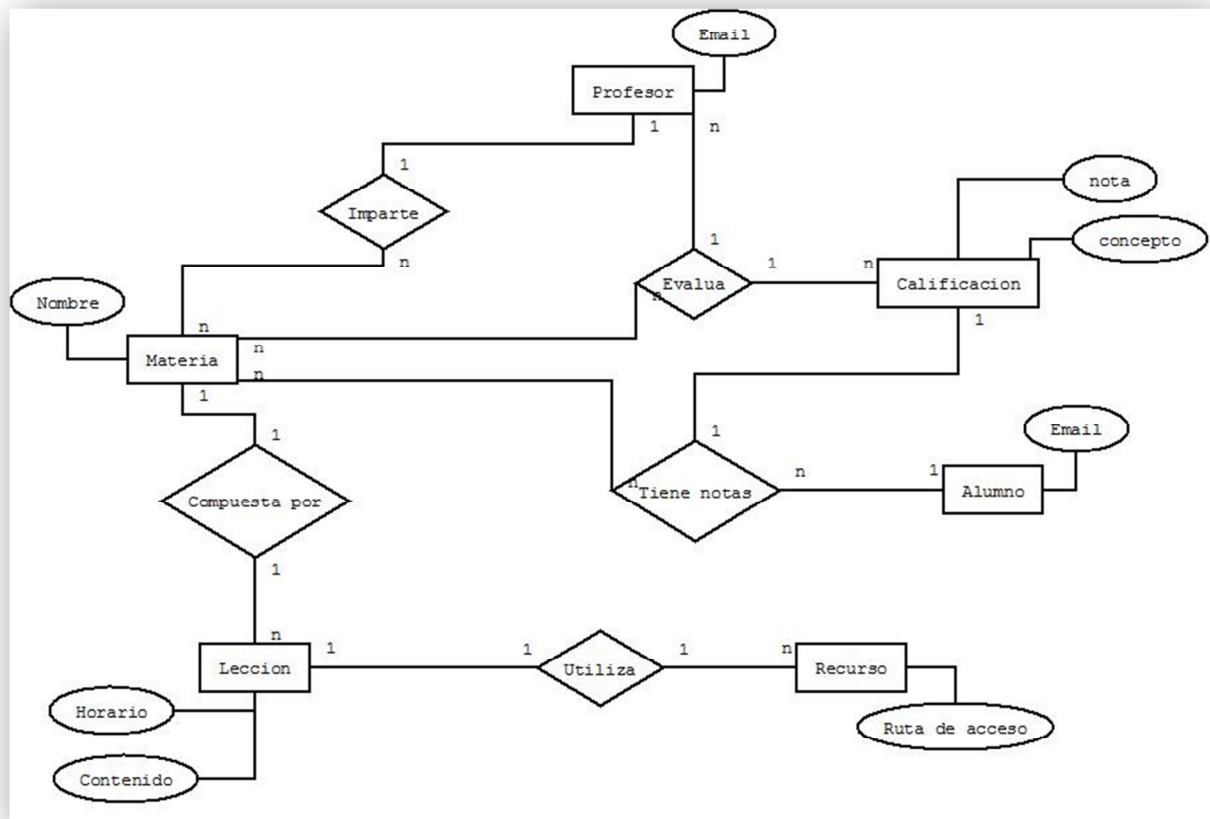


Ilustración 9: Modelo E/R

Las entidades principales del modelo tienen que ver con el universo del E-Learning y han sido muy condicionadas por los requisitos del sistema. Este modelo da una idea muy cercana de los conceptos de negocio que se están manejando, así como las funcionalidades que pueden implementarse desde los mismos.

Así, tenemos la entidad profesor que se encarga de impartir una o más materias y de evaluarlas para generar una o más calificaciones para los alumnos. Asimismo, la materia está compuesta por lecciones que utilizan recursos. Un alumno tiene notas de una o más materias.

4.2.2. Modelo Relacional

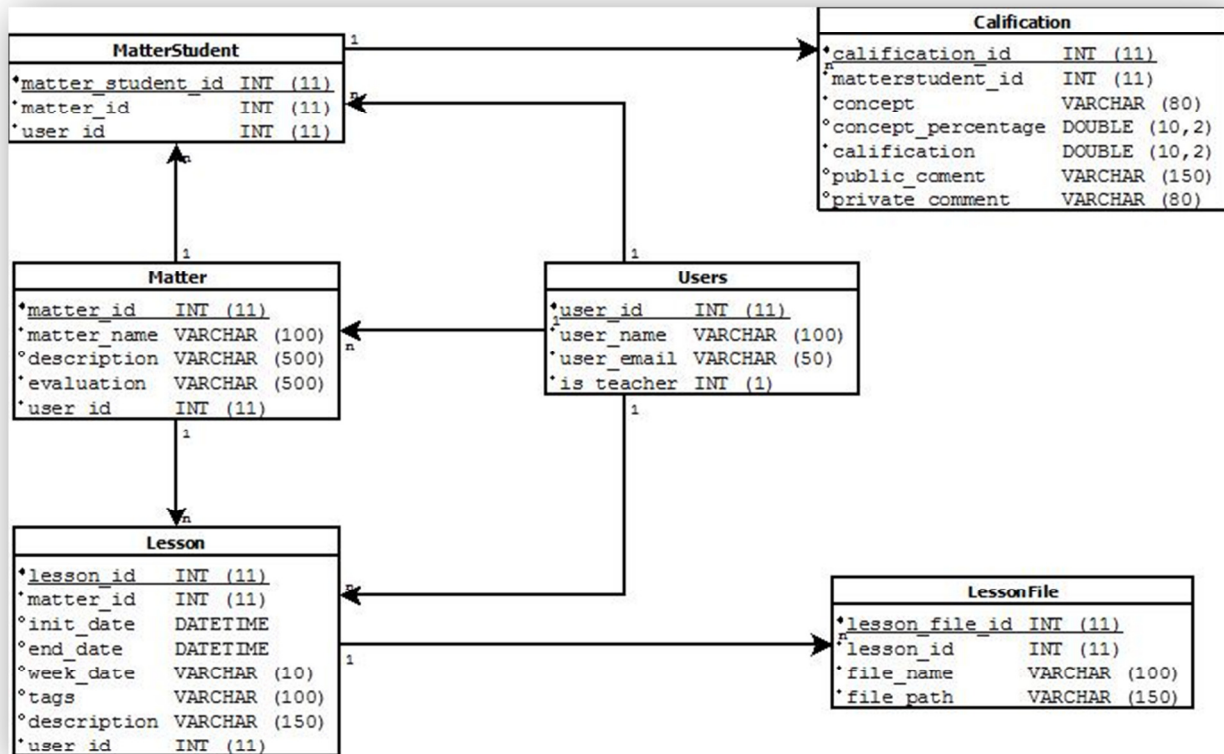


Ilustración 10: Modelo Relacional

Se han tenido en cuenta las principales entidades del modelo anterior para generar las tablas en la base de datos con los principales atributos característicos de los mismos, con la particularidad de que se ha unificado el concepto profesor y alumno, que tenían como principal característica el email, para crear la tabla users, mucho más polivalente a nivel aplicativo, sobre todo a la hora de gestionar los logins.

Así, tenemos una tabla Matter que representa las asignaturas del aplicativo que está relacionada directamente con la tabla Users para establecer el profesor de la asignatura. Así mismo, se relaciona a través de la tabla MatterStudent (también relacionada con la tabla Users) para identificar a los alumnos suscritos a la misma. Cada alumno tendrá una o varias calificaciones en la tabla Calification.

Por otro lado, cada asignatura está compuesta por lecciones representadas en la tabla Lesson. A su vez, cada lección, podrá tener n recursos almacenados en la tabla LessonFile.

4.3. Vista de Implementación

A continuación desgranaremos cada componente que conforma la infraestructura para entender los propósitos y la necesidad de cada uno de ellos.

4.3.1. Vista

Empezaremos con la vista, dado que contendrá la interfaz de entrada y de salida de la aplicación con el usuario. Al ser una plataforma Web, está orientada a ser visualizada desde navegadores web. Se compondrá de una serie de páginas y formularios que estructurarán la información que se muestra y permitirá ejecutar acciones que afectarán a los datos, como por ejemplo: editarlos, añadirlos o eliminarlos.

Para facilitarnos esta tarea, hemos optado por implementar una arquitectura MVC en este componente, dado que se antoja fundamental manejar correctamente los datos que se recogen del usuario, puesto que es responsabilidad de este componente validarlos y asegurarnos de que responden a las necesidades esperadas por el modelo. Por ejemplo, una dirección de email debe incluir una arroba y un campo numérico no debe incluir caracteres de texto. También deberá mostrar los datos correctamente al usuario, por lo que es necesario trabajar sobre ellos antes de mostrarlos, ya sea para transformarlos a entes entendibles en los diversos contextos, o simplemente para estructurarlos de manera adecuada para su fácil visualización y comprensión.

Existe una tecnología ampliamente extendida para ayudarnos a conseguir todo esto y que facilitará la gestión de la arquitectura de este componente orientado a Web. Se trata de Struts.

Struts.

Struts es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma Java EE. Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto independiente conocido como Apache Struts.

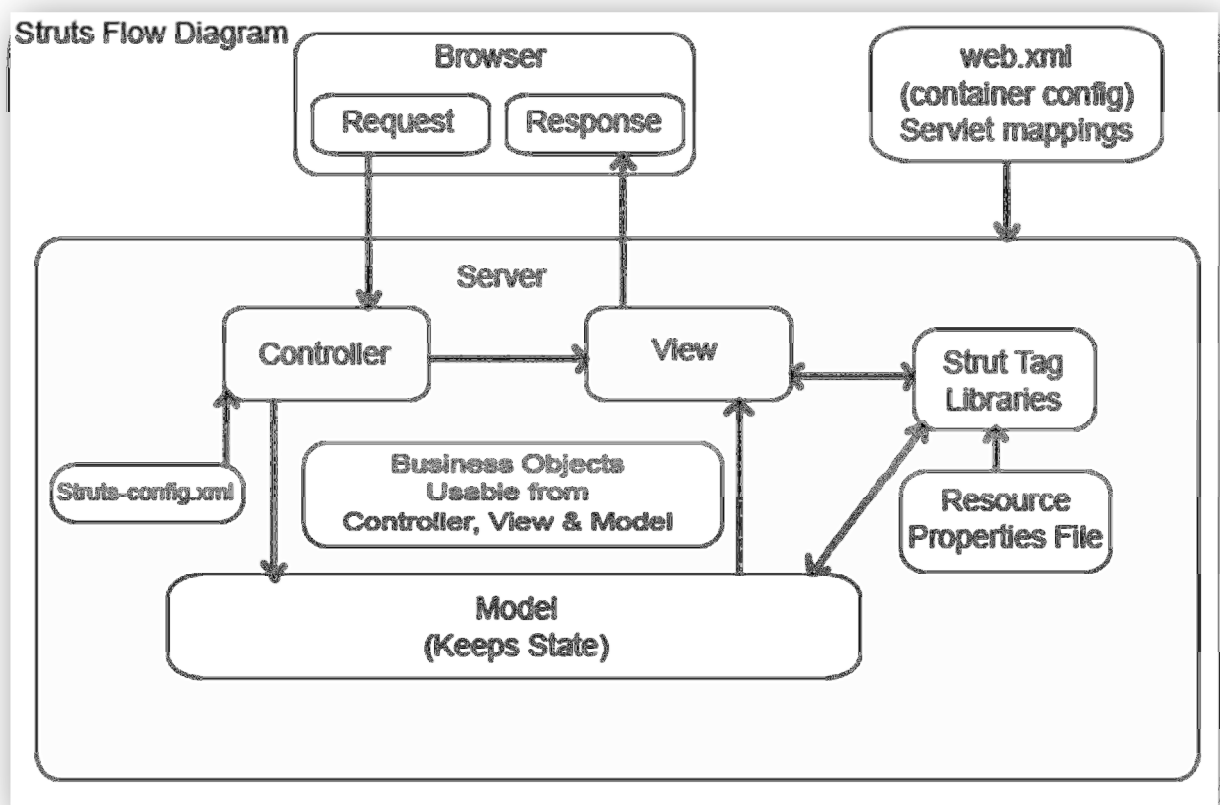


Ilustración 11: Struts

Para implementar la arquitectura propuesta por Struts, usaremos 3 tipos de estructuras correspondientes a las capas del modelo MVC. Además, tendremos XML de configuración propia de la tecnología, como el Struts-Config, y otros ajenos como el SpringContext, que explicaremos más adelante.

- **JSP:** Compondrá la Vista de la arquitectura. Serán las encargadas de generar el html interpretado por los navegadores. Harán uso de los recursos que aporta Struts para

la iteración de listas, recuperación y validación de datos, envío de formularios, información de errores, etc. Se apoyará en tecnología css para definir los formatos de las páginas y contendrán los scripts necesarios para el funcionamiento dinámico de la página (hemos usado JavaScript para este fin).

- ✚ **Form:** Será el modelo de la arquitectura. Se trata de objetos con atributos simples que pueden ser recuperados y reemplazados mediante getters y setters. Representan el modelo de datos de la aplicación y según la configuración de Struts aplicada, serán los responsables de la validación estructural de los datos.
- ✚ **Action:** Es el controlador de la Vista. Se encarga de gestionar las acciones ejecutadas por el usuario en la interfaz y de llamar a las otras capas de la arquitectura general para obtener un servicio (ya sea de almacenaje, de información o de transformación). Se encarga de asegurar que los datos estén disponibles para la visualización en las interfaces de usuario y gestiona la navegación entre una y otra.

4.3.2. Controlador

La capa controlador se encarga de hacer de intermediario entre el modelo y la vista. Traducirá las peticiones que hacen los usuarios a través de la interfaz a mensajes entendibles por el modelo y viceversa.

A través de interfaces, esta capa ofrece a la Vista una serie de servicios que permiten interactuar con datos y con otros sistemas. Estos servicios serán invocados por los Actions implementados en Struts, que son los que realizan la labor de controlador en la subestructura MVC de la Vista; por esta razón estos dos componentes estarán muy relacionados. Sin embargo, el uso de interfaces desde el controlador, permite abstraer la implementación de estos servicios, desacoplando una capa de otra, dado que desde la vista no se tiene constancia de cómo se ejecutan sus peticiones. Además, en este punto debemos mencionar el uso de la tecnología Spring, que entre otras cosas, permite centralizar la configuración de interfaces y sus implementaciones. Así, usando además polimorfismo, conseguimos que resulte muy fácil usar una u otra implementación de una interfaz según nuestras necesidades.

Por otra parte, el controlador también se encarga de llamar a las interfaces que pone a su disposición el Modelo, del que recupera, añade y edita información. Para conseguir esto, primero debe haber una traducción en los datos que llegan desde la vista, para que se adecuen a las APIs que ofrece el Modelo y puedan ser entendidas. Del mismo modo, se hace

el proceso inverso: traducir las respuestas que obtenemos del Modelo para enviarle a las Vista la información que necesita.

Spring

Spring, es un framework para el desarrollo de aplicaciones que facilita la centralización y organización de la encapsulación del sistema. Así, puede usarse para ayudarnos a implementar un modelo MVC haciendo uso de su inyección de dependencias, que junto al uso de interfaces y de polimorfismo, da al sistema una gran robustez y maleabilidad. La configuración del sistema se ejerce desde uno o varios XML usualmente denominados SpringContext. Algunas de las características de Spring que se usarán en el proyecto son:

- ❖ **Singleton:** Es un patrón de diseño que sirve para garantizar una única instancia por objeto, de forma que ahorremos memoria y se eviten problemas de inconsistencia. Usando Spring como sistema centralizado de beans, nos aseguramos de disponer siempre de la misma instancia.
- ❖ **Polimorfismo:** Usando Spring, se puede trabajar únicamente con interfaces, dado que no es necesario instanciar específicamente ninguna clase, dado que este framework se encarga de ello en tu lugar. Configurando para cada caso la instancia que tendrá cada interfaz desde un lugar centralizado, resulta sencillo modificar la implementación del sistema de forma desacoplada.
- ❖ **Inyección de dependencias:** Se puede instanciar una clase mediante la inclusión de parámetros definidos en Spring, de esta manera independizamos aún más la instanciación de la implementación, pasando en cada caso las dependencias necesarias.

4.3.3. Modelo

Es la capa que contiene y gestiona el modelo de datos del sistema. Ofrece al controlador una interfaz a través de la que acceder a los datos del mismo, que esta capa se encarga de organizar en entidades simples y manejables. Así mismo, se encarga de usar los servicios de la infraestructura para acceder y almacenar los datos en los sistemas finales. Consideraremos modelo a tres tipos de clases:

- ❖ **Entity:** Son los objetos del modelo y están muy relacionados con la estructura de la base de datos. Están formados únicamente por atributos con getters y setters estándar, tal y como ocurría con los Forms de la capa Vista. Con la ayuda de Spring y de XMLs que definan el modelo de datos, es posible parsear directamente la información recuperada por los daos de la base de datos a los propios objetos de la aplicación. Esta utilidad aligerara la gestión de los datos automatizando el proceso.
- ❖ **Dao:** Son las clases que se encargan de recuperar y almacenar los datos con los que rellenar las Entitys de información. Contendrán la lógica necesaria para acceder a los datos, pero no implementarán el acceso, dado que de esto se encarga la capa de infraestructura que explicaremos después. Se ofrecerá una interfaz de estos métodos al controlador, de forma que esta capa pueda recuperar los objetos necesarios para su posterior traducción para la vista.
- ❖ **Entrada y salida (E/S):** Tiene una función similar a los Daos, pero estos están más orientados a ficheros que a un modelo de base de datos. Ofrecerán interfaces para que el controlador pueda acceder y almacenar ficheros en distintos formatos, aunque no se encargará específicamente de la implementación, dado que se delegará esta función a la capa de infraestructura, dado que se utilizarán las APIs que ofrecen las tecnologías que usaremos.

Además del modelo propio, también consideraremos otros servicios ofrecidos externamente pero que por su importancia, no pueden ser ignorados. Así, para la gestión de los usuarios y la seguridad se ha utilizado el login de Facebook, por lo que consideramos que esta aplicación se sitúa en esta capa de la aplicación. El Controlador se encargará de llamar a su API, traduciendo los datos para que puedan ser entendidos por sus interfaces. Los datos recuperados serán gestionados por las clases del modelo y posteriormente podrán ser almacenados y consultados por la capa de infraestructura.

También se usa una librería externa para la generación de ICS (calendarios), que según los requisitos del sistema, deberá ser capaz de generar estos ficheros. Al igual que ocurría con Facebook, el controlador se encargará de usar su API y el modelo gestionará los datos que se obtengan, almacenándose posteriormente por la Infraestructura.

4.3.4. Infraestructura

Esta capa puede o no independizarse de la del modelo. En nuestro caso hemos decidido independizarla dado que nos apoyaremos en tecnologías como Hibernate para abstraernos del funcionamiento de la base de datos. Tendremos varios tipos de herramientas para ayudarnos a implementar esta capa que organizaremos según estén orientadas a ficheros o a bbdd:

- ❖ **Base de Datos (BBDD):** Para gestionar el acceso a los datos almacenados en las tablas del modelo y garantizar la integridad de los mismos se delega la inserción, actualización y recuperación de datos a la tecnología Hibernate. Hibernate se encarga de la gestión de las conexiones a la base de datos y ofrece una interfaz con métodos para trabajar sobre la misma, que será usada por la capa del modelo. De esta forma, nos olvidamos de la implementación y del control del acceso a la base de datos, delegando esta tarea a una tecnología ampliamente conocida y probada.
- ❖ **Ficheros:** Para gestionar el acceso y almacenaje de ficheros, hemos usado la API de Dropbox. Dropbox se encargará de almacenar y de recuperar los ficheros y sus metadatos de la nube, por lo que el Modelo solo debe encargarse de realizar las llamadas que Dropbox ofrece en su interfaz. De esta forma, nos olvidamos de la implementación de la E/S de estos ficheros, delegando la gestión de conexiones a esta tecnología. Así también aumentamos la seguridad de la misma.
- ❖ Puesto que también es necesario generar ficheros que no serán almacenados en la nube, también pertenece a la capa de infraestructura una librería encargada de generar ficheros ICS (de calendario). Que tendrá la capacidad de realizar E/S para almacenar estos ficheros en el disco duro.

4.4. Vista de Ejecución

El objetivo de este apartado será mostrar cómo interactúan los distintos elementos desgranados en la vista de implementación, para entender cómo se comporta la aplicación cuando se genera un evento por parte del usuario. En el siguiente diagrama de flujo, tratamos de mostrar desde un punto de vista arquitectónico, el recorrido que realizará una petición de usuario desde el navegador, pasando por todas las capas de la infraestructura y la labor que ejerce cada componente en la misma.

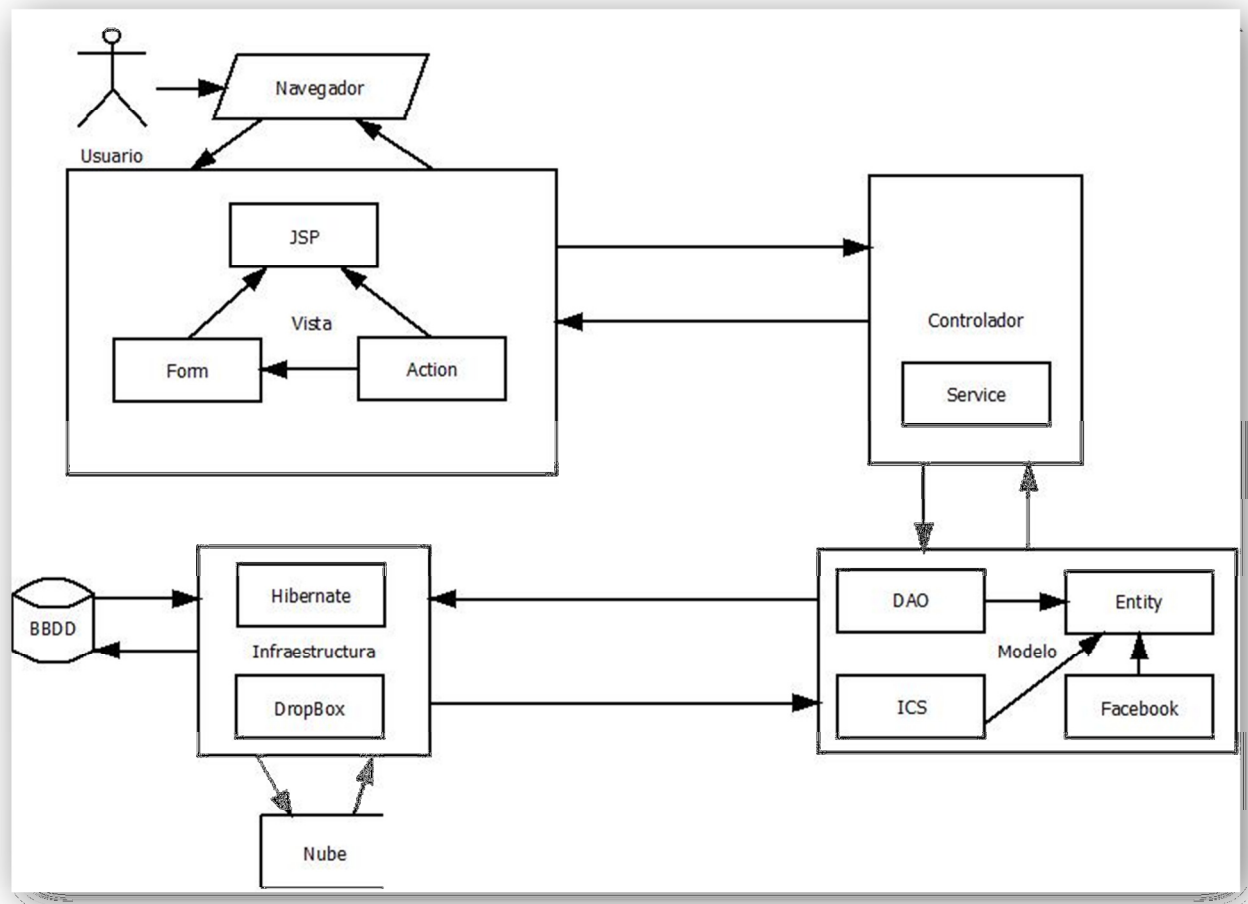


Ilustración 12: Diagrama de Flujo

Como puede verse, el usuario interactuará con el navegador, el cual realizará peticiones al servidor sobre el que tendremos montado el sistema. La primera capa a la que accederemos será la vista, que esta subcompuesta en un modelo MVC implementado con la ayuda de Struts. En esta arquitectura, el mensaje se recogerá en los Actions, que realizará peticiones a la segunda capa de la aplicación, el controlador. En el controlador facilitaremos interfaces de servicios que se encargaran de traducir el mensaje desde la capa de Vista para que pueda ser entendido por las interfaces del Modelo. De esta forma, si se desea consultar la base de datos, los servicios llamarán a la interfaz de los Dao para que a su vez estos hagan la petición a Hibernate, que será el componente que recupere la información de la bbdd en la cuarta capa de la arquitectura. De vuelta al modelo, los Dao almacenaran la información recuperada en las Entitys, que se enviarán a los servicios y a su vez a la vista. Los action traducirán la información del modelo de datos de la aplicación al modelo de datos de los Forms y definirán como se muestra la información por las JSP. El navegador imprimirá el html generado por la JSP y actuará como interfaz de salida para el usuario.

Una vez entendido el comportamiento general del sistema, vamos a ejemplarizar varias acciones típicas en diagramas de secuencia. Con ellos podremos ver el recorrido que hacen los mensajes en 3 situaciones y las respuestas del sistema.

4.5. Vista de Despliegue

La vista de despliegue tiene como objetivo mostrar los componentes hardware del sistema. Así, para que la plataforma funcione, se requiere un servidor donde instalar el software y que contendrá todos los componentes del sistema, excepto la base de datos, que puede estar o no desplegada en la misma máquina; así como los dos elementos externos de la aplicación: Dropbox y Facebook. Por otra parte, los usuarios dispondrán de dispositivos, que pueden ser móviles, tablets, PCs, o en definitiva, cualquier soporte que admita un navegador compatible con la aplicación.

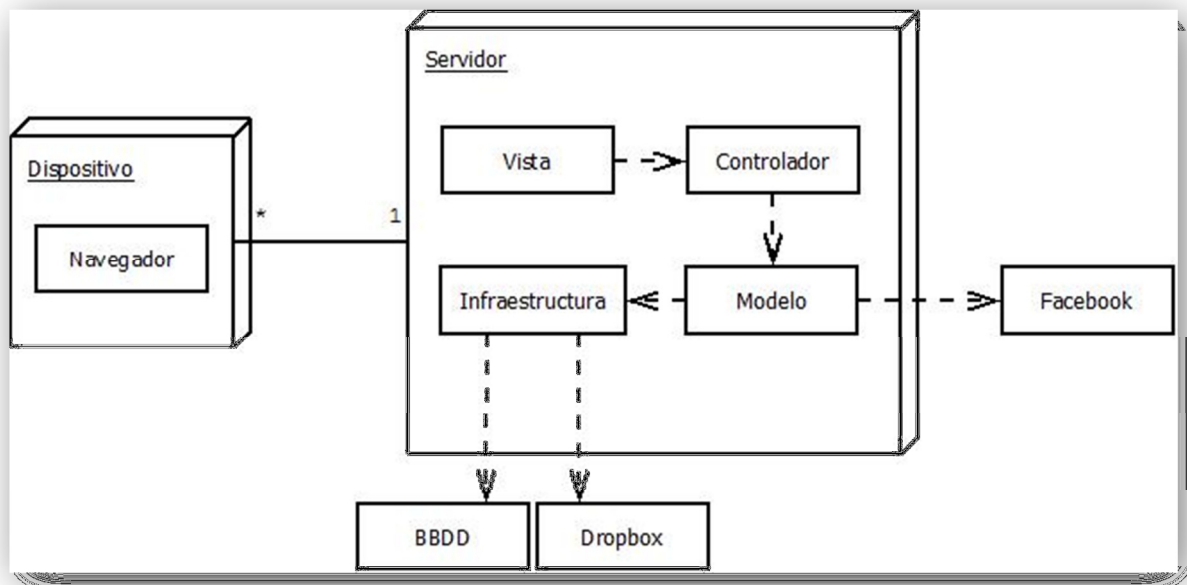


Ilustración 13: Vista de Despliegue

4.6. Alternativas de diseño

En la actualidad, existen múltiples tecnologías sobre las que diseñar una aplicación de estas características. En muchas ocasiones, la elección de una u otra depende simplemente de la experiencia trabajando con una u otra alternativa, dado que no es raro que sus características y prestaciones sean similares. Pasamos a describir algunas posibles alternativas a las tecnologías que finalmente se han seleccionado:

- ❖ **.NET**: Es la plataforma de desarrollo de Microsoft. Ofrece un entorno simple y amigable para el desarrollo web, pero es difícil encontrar Frameworks adicionales para resolver diversos aspectos como ocurre en Java.
- ❖ **Grails**: Utiliza Groovy como lenguaje de programación. Este lenguaje soporta la sintaxis y las librerías de Java, además de ampliarlo con nuevas funciones y potentes variaciones. Se critica la falta de formalidad del lenguaje, dado que puede omitirse gran parte del léxico estructurador propio de Java, lo que puede hacerlo confuso.
- ❖ **JBoss**: es un servidor de aplicaciones que puede ser una alternativa más potente que Tomcat, pero en general es más difícil de configurar y de montar.
- ❖ **Oracle**: Es un servidor de base de datos alternativo a MySQL. Suele ser usado en entornos de producción de grandes empresas debido a todas las opciones que ofrece. Su principal desventaja es que muchas de sus características no son gratuitas.
- ❖ **ANT**: Es la alternativa de Maven, más apropiada para proyectos pequeños puesto que desarrolla su actividad en un Script. Carece de la potencia de Maven para controlar las dependencias transitivas.
- ❖ **EJB**: Es la alternativa de Spring. Ofrece unas prestaciones muy similares, por lo que la elección de uno u otro se debe al dominio que se haya adquirido en cada uno de ellos.
- ❖ **Eclipse Link**: Es una posible alternativa a Hibernate como método de implementación de las interfaces Java: JPA y JDBC.

4.7. Diseño detallado

Para el diseño detallado vamos a desglosar la plataforma desde un punto de vista funcional. Así, cada pantalla de la aplicación se mostrará con sus dependencias y diagrama UML relacionado. Se explicarán funcionalmente los botones y las acciones que se pueden realizar.

Este es el diagrama de pantallas de la aplicación, con su navegabilidad entre una página y otra. El usuario se identificará en la página de Login. Si el usuario intenta acceder a cualquier otra página sin estar autenticado, se le redireccionará automáticamente a esta página para que se valide como usuario. Una vez validado, podrá acceder a dos vistas principales: la del profesor y la de la lección.

- ❖ En la vista del profesor podrá acceder a las páginas de creación y edición de cronogramas. También podrá acceder a la vista de lección.
- ❖ Desde la vista de lección se podrá acceder a la página de suscripción de cursos, desde la cual se podrá acceder a la vista del profesor. Además, desde la vista de lección se podrá visualizar la página de evaluaciones.

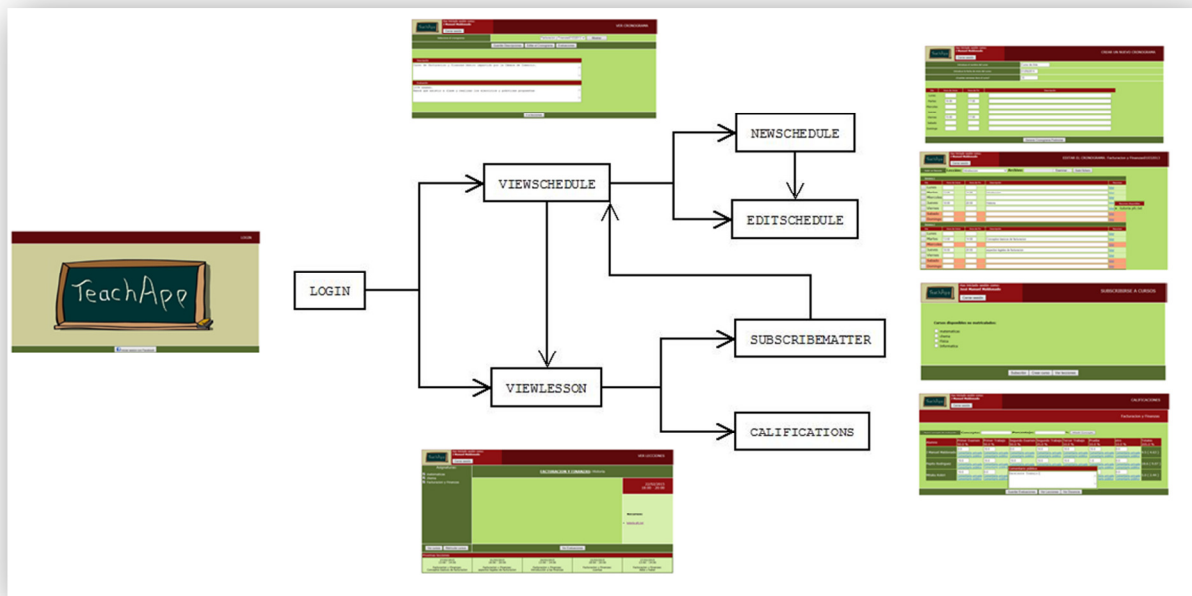


Ilustración 14: Diagrama de Pantallas

4.7.1. Login

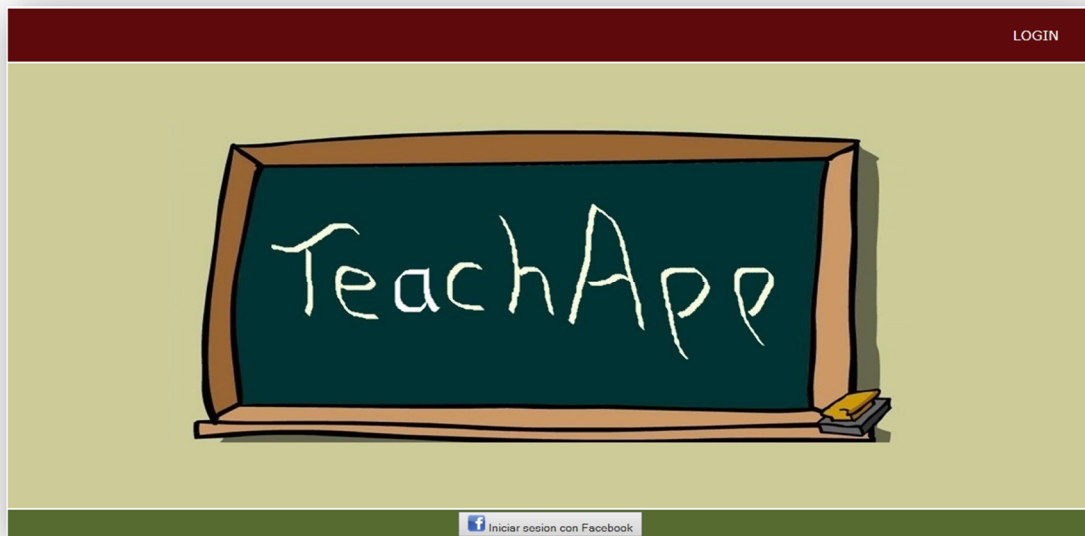


Ilustración 16: Login

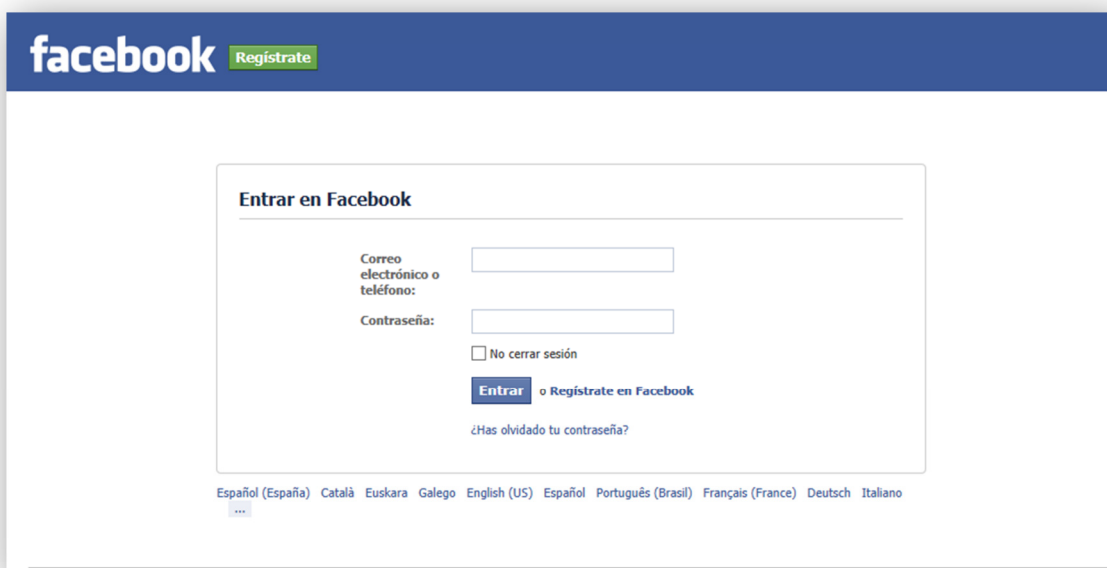


Ilustración 15: Login Facebook

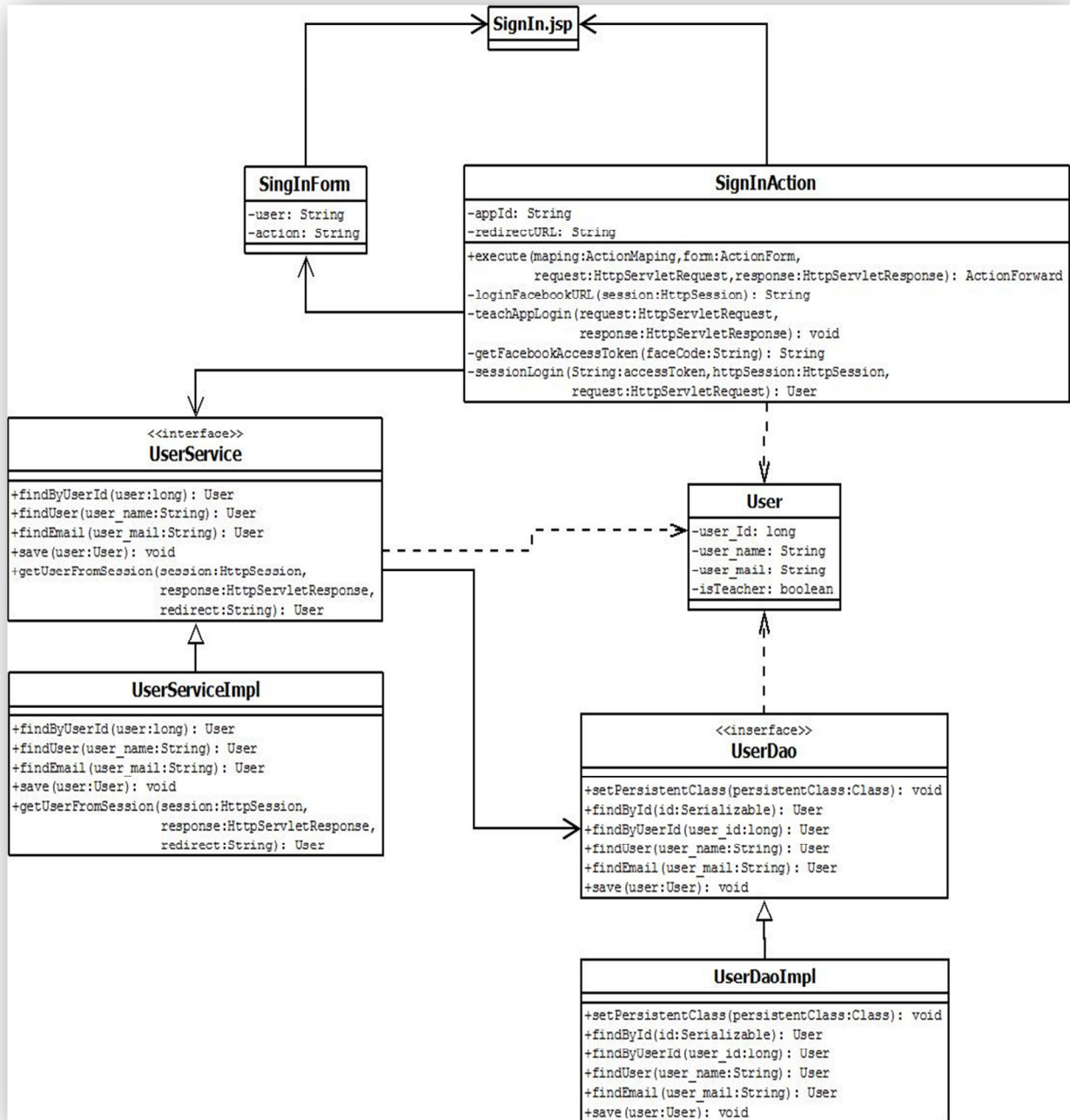


Ilustración 17: UML login

La aplicación redireccionará a esta página siempre que en la sesión del navegador que se esté utilizando no se encuentre logado. Esto se comprobará a través del método `getUserFromSession` del `UserService`. Este método extraerá el mail de la sesión del usuario para buscarlo en bbdd y extraer toda su información relativa. En caso de que el usuario no este logado, se redireccionará a la página de login, siempre con una referencia a la URL de la cual se proviene. En esta página habrá un botón que el usuario pulsará para redireccionar a Facebook. Desde la interfaz facilitada por Facebook se iniciará sesión ingresando el mail y la contraseña de esta red social. Una vez comprobado por Facebook la validez de los datos, se redireccionará a la página inicial a la que estaba tratando de acceder el usuario, ya con los datos del usuario en la sesión. Esta operación se llevará a cabo desde cualquier página de la aplicación, pero por motivos de claridad, no se mencionará en el resto del diseño.

En el caso de que el mail del usuario que haya iniciado sesión en Facebook no haya sido registrado aún en base de datos, se procederá a ingresar una tupla con sus datos y se le asignará un id: el mail, el nombre y si es profesor o no.

El servicio debe implementar métodos para buscar un usuario en bbdd por mail, por id y por nombre.

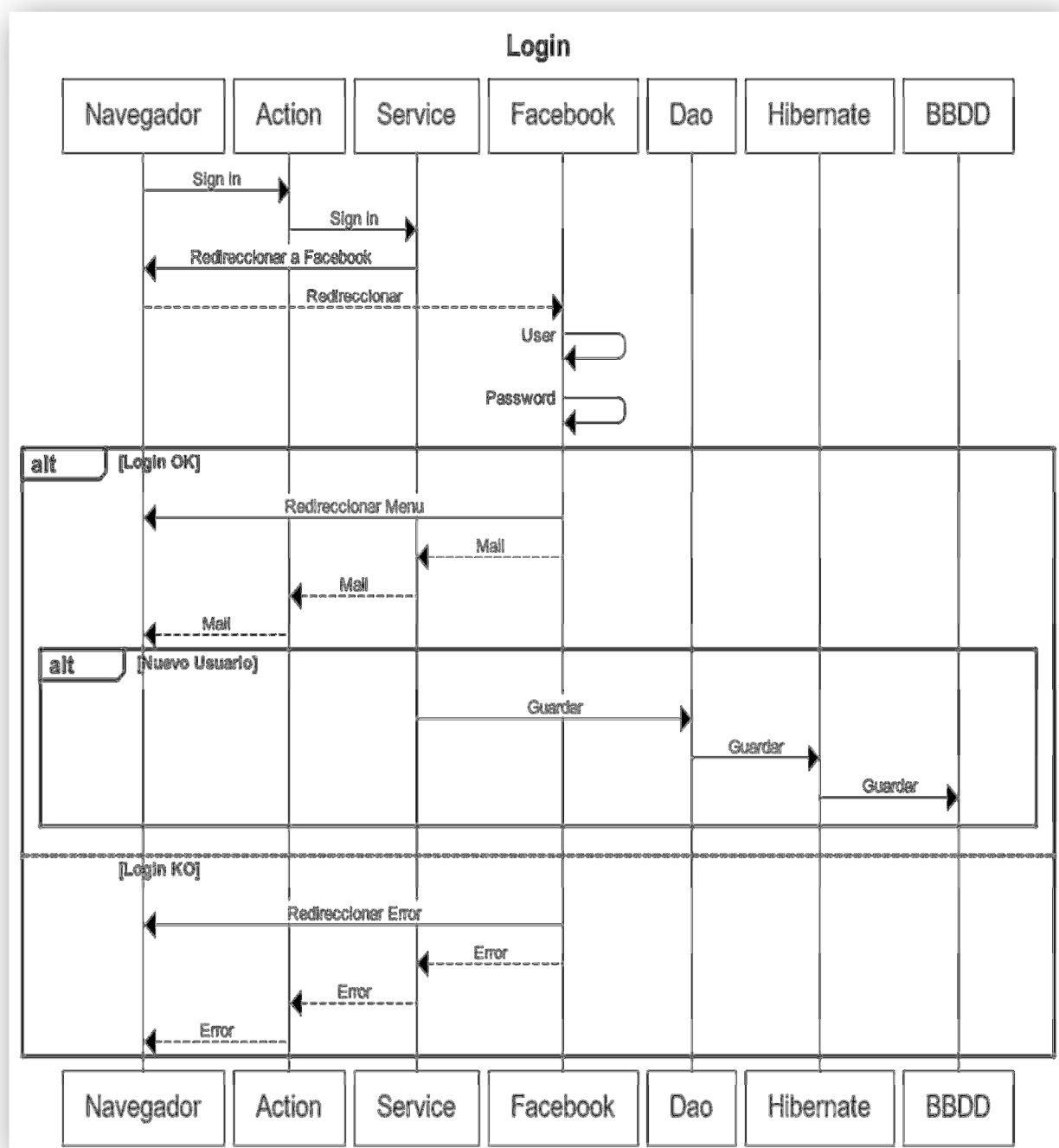


Ilustración 18: Diagrama de secuencia Login

El usuario realiza la petición de login en el navegador. El action se encarga de recoger esta petición e invoca al servicio de login, que indica al navegador que debe redireccionar la página a una URL de Facebook. Facebook se encargará de ofrecer una interfaz donde el usuario podrá ingresar su identificador y su contraseña. Si la identificación es errónea se devolverá error y se redireccionará a una página de error. Si la autenticación es correcta se redirecciona al menú de la aplicación y se comprobará si el usuario está registrado en la base de datos. Si no lo está, se lanza desde el servicio una petición de guardado a los Daos. Este a través de Hibernate se encargará de registrar al usuario en la base de datos.

4.7.2. Vista Lección

The screenshot shows the TeachApp interface. At the top, a dark red header bar contains the 'TeachApp' logo, a user login status 'Has iniciado sesión como: J Manuel Maldonado', a 'Cerrar sesión' button, and a 'VER LECCIONES' link. Below the header, the main content area is divided into a sidebar on the left and a main panel. The sidebar, titled 'Asignaturas:', lists 'matematicas', 'chema', and 'Facturacion y Finanzas', all of which are checked. The main panel is titled 'FACTURACION Y FINANZAS: Historia' and features a large green rectangular area. To the right of this area, a red box displays the date and time '22/02/2015 18:00 - 20:00', and a 'Recursos:' section lists a file 'tutoria.pfc.txt'. Below the main panel, there are buttons for 'Ver cursos', 'Matricular cursos', and 'Ver Evaluaciones'. At the bottom, a 'Proximas lecciones' section contains a table of upcoming lessons.

Proximas lecciones				
27/02/2015 13:00 - 14:00	01/03/2015 18:00 - 20:00	20/03/2015 13:00 - 14:00	22/03/2015 18:00 - 20:00	27/03/2015 13:00 - 14:00
Facturacion y Finanzas: Conceptos basicos de facturacion	Facturacion y Finanzas: aspectos legales de facturacion	Facturacion y Finanzas: Introducción a las finanzas	Facturacion y Finanzas: cuentas	Facturacion y Finanzas: debe y haber

Ilustración 19: Vista lección

Esta será la página principal del alumno. En ella podrá visualizar las asignaturas que tiene suscritas y seleccionar aquellas sobre las que quiere visualizar información. En el cuerpo de la página aparecerá la próxima lección programada en el tiempo de las asignaturas seleccionadas y se ofrecerá un breve resumen de las características de la asignatura. Además se incluirá información específica de la lección, como la hora a la que se impartirá, el temario previsto y los recursos docentes que se facilitan. Habrá un tercer cuadro para navegar por las próximas lecciones a las que está suscrito el alumno, que se ordenaran cronológicamente; en ellos se mostrará el horario de la lección, así como la asignatura y el temario previsto.

Esta página permitirá acceder a diversas funciones:

- ❖ **Ver cursos:** permitirá seleccionar de la lista de los cursos suscritos aquellos para los cuales se desea visualizar información, obviando el resto.
- ❖ **Matricular Cursos:** Redirigirá a una página en la cual podemos ver un listado de los cursos ofertados para la matriculación de los mismo, de forma que aparezcan en el listado de asignaturas suscritas del alumno.
- ❖ **Evaluaciones:** Redirigirá a una página en la cual podemos visualizar las calificaciones obtenidas por el alumno en las distintas asignaturas matriculadas, así como los comentarios especificados por el profesor para cada evaluación.



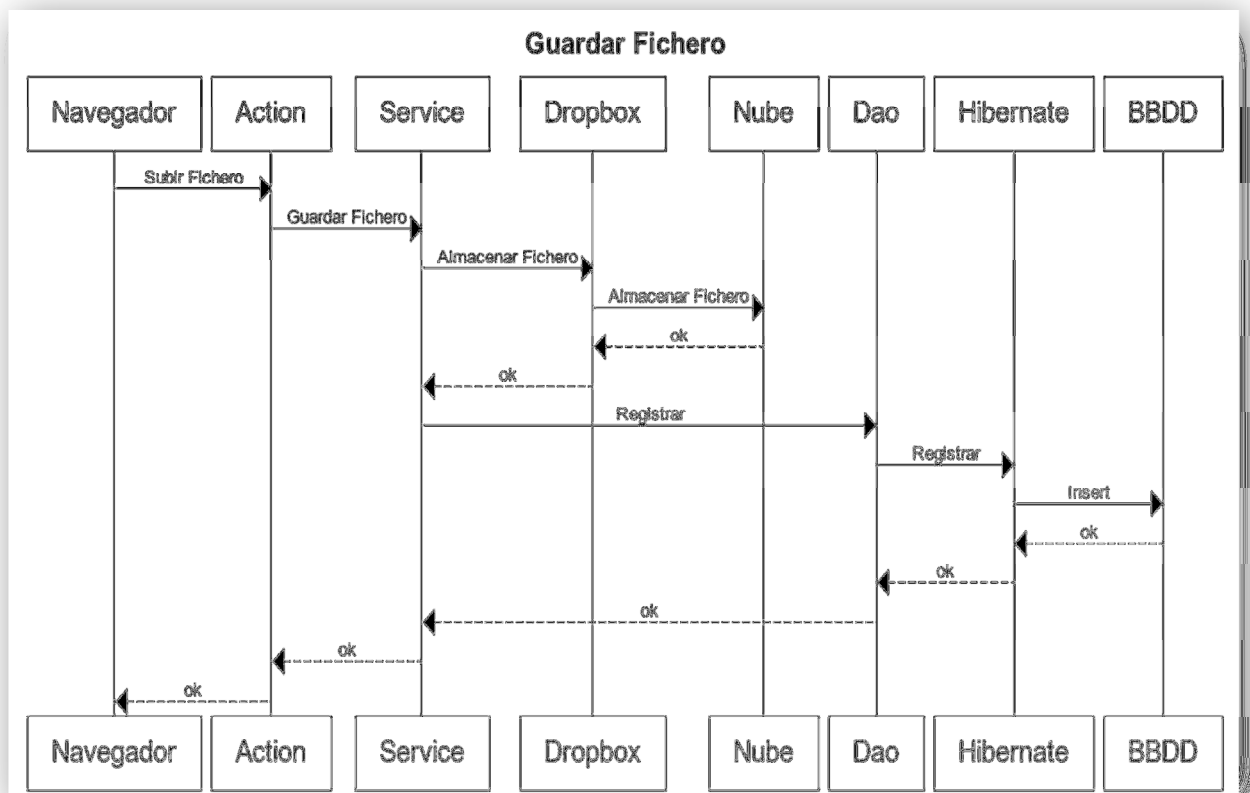


Ilustración 21: Diagrama de secuencia guardar Fichero

Desde el navegador el usuario podrá seleccionar un fichero de su disco duro para subirlo a la nube. Esta petición será recogida por un action, que invocará el servicio de la aplicación. Este servicio usará la API de Dropbox para subir el fichero seleccionado por el usuario a la nube. Si la petición es ok, se registrará este hecho en la base de datos. Para ello el servicio llamará al dao, y este realizará el almacenaje de la información en la base de datos a través de Hibernate. Una vez hecho todo esto, se devolverá un mensaje de OK al navegador para indicar que la acción se ha llevado a cabo correctamente.

4.7.3. Matriculación De Cursos



Ilustración 22: Vista Matriculación de cursos

En esta página el usuario podrá matricularse de las asignaturas ofertadas en la plataforma. Para ello se mostrará un listado de las asignaturas a las cuales todavía no se ha suscrito y le permitirá seleccionarlás. Pulsando el botón 'suscribir' el usuario quedará suscrito a las asignaturas.

También existirá la opción de crear un curso. Para ello se pulsará el botón de crear curso y se redireccionará a la página de profesor, desde donde podrá crear un nuevo cronograma y publicar contenido en la asignatura para que pueda ser matriculada por otros alumnos. Al hacer esto, el usuario pasa a tener perfil de profesor para las asignaturas creadas, pero no perderá su estatus de alumno para el resto.

Existirá un tercer botón 'volver' para redirigir a la página principal del alumno: vista lección.

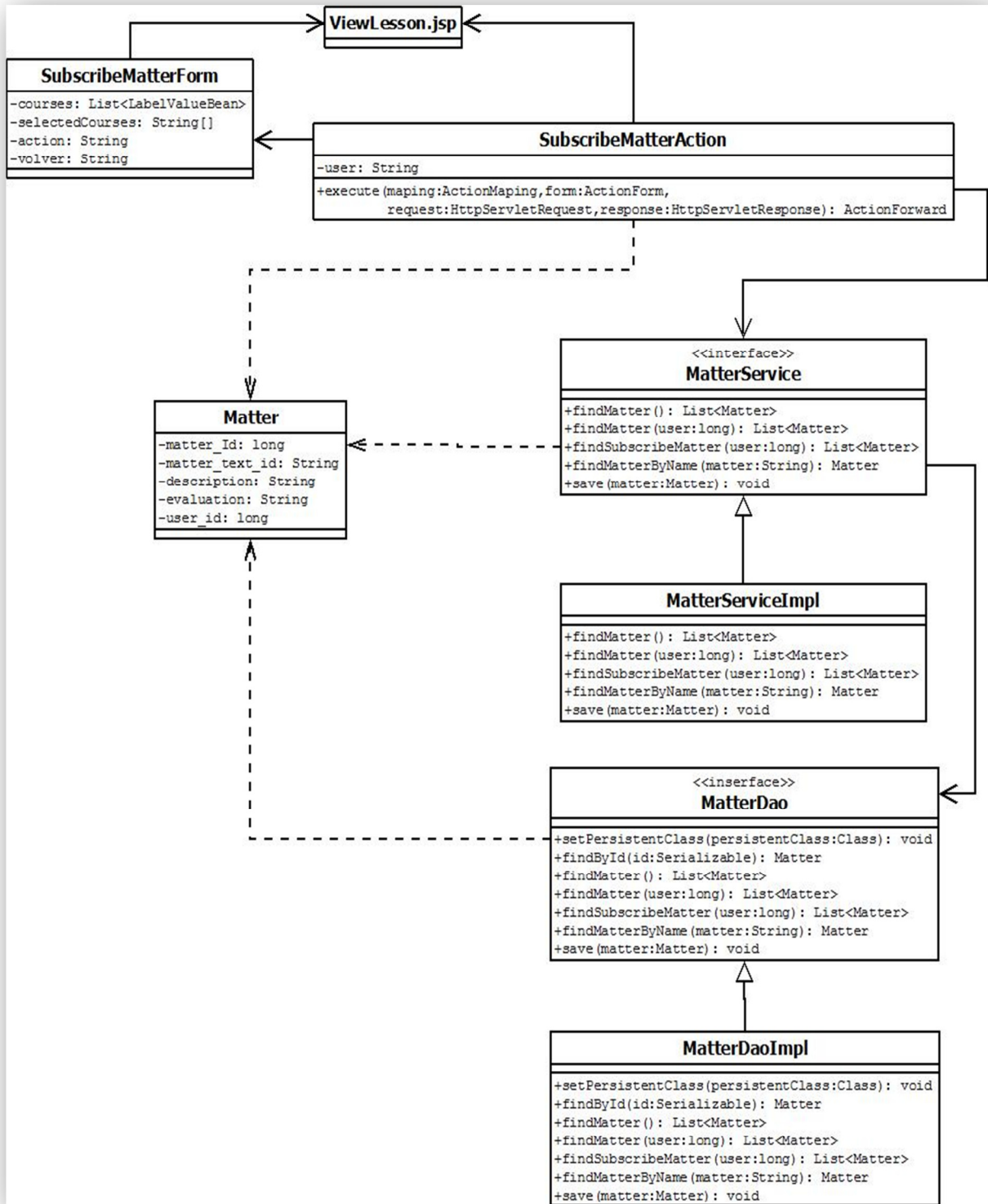


Ilustración 23: UML matriculación de cursos

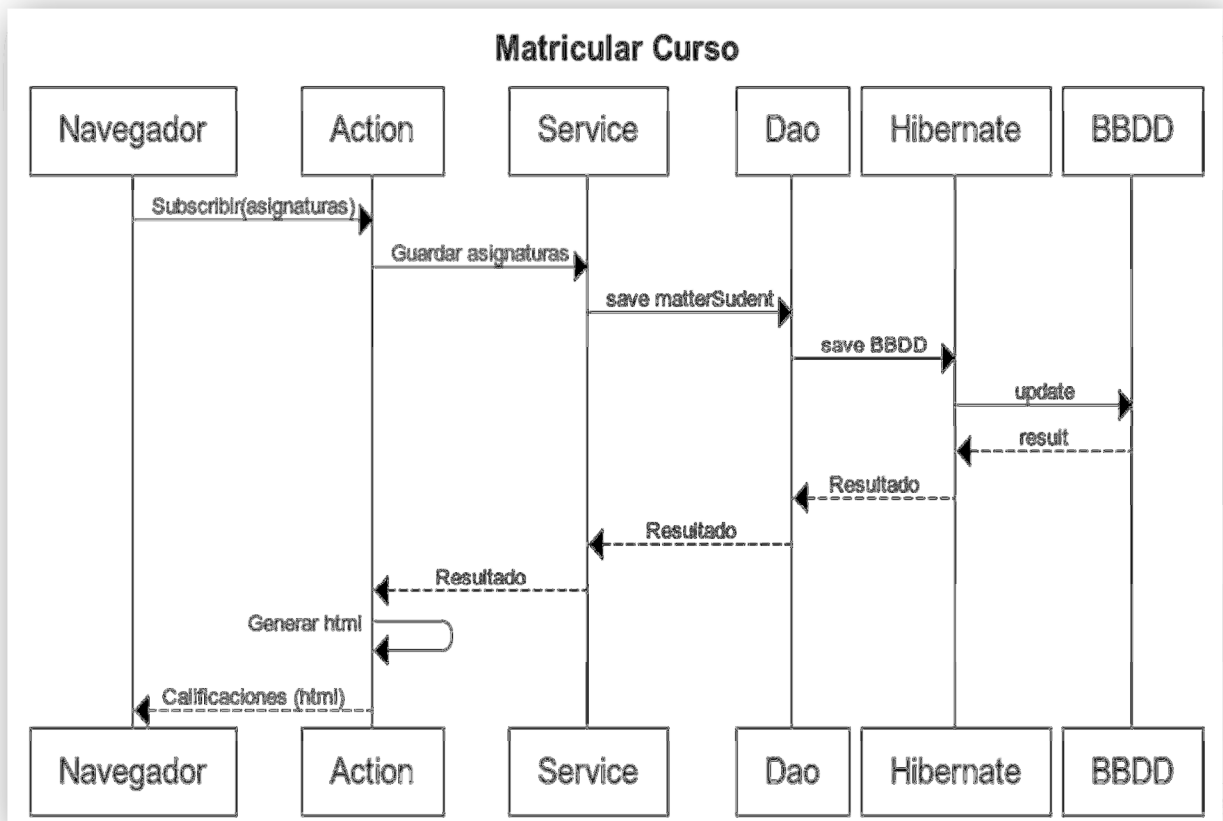


Ilustración 24: Diagrama de secuencias de matriculación de curso

Desde el navegador se seleccionan los cursos que se deseen matricular de los ofertados por la aplicación. Al pulsar el botón subscribir se envía esta información al action, que a su vez llamará al servicio que gestiona la subscripción de asignaturas. Este servicio llamará al Dao de la tabla matterStudent, que relaciona las asignaturas con los usuarios y este solicitará la inserción o actualización del registro en bbdd a través de la interfaz de Hibernate, retornando en cascada el resultado hasta el navegador.

4.7.4. Calificaciones

Has iniciado sesión como: **J Manuel Maldonado**

Cerrar sesión

CALIFICACIONES

Facturación y Finanzas

Nuevo concepto de evaluación **Concepto:** **Porcentaje:** %

Alumno	Primer Examen 50.0 %	Primer Trabajo 50.0 %	Segundo Examen 50.0 %	Segundo Trabajo 25.0 %	Tercer Trabajo 10.0 %	Prueba 10.0 %	otro 10.0 %	Totales 205.0 %
J Manuel Maldonado	0.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	0.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	0.0 Comentario privado Comentario público	9.5 (4.63)
Pepito Rodriguez	10.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	1.0 Comentario privado Comentario público	0.0 Comentario privado Comentario público	18.6 (9.07)
Mitaku Kokiri	10.0 Comentario privado Comentario público	0.0 Comentario privado Comentario público	Excelente Trabajo: <input type="text"/>			Comentario privado Comentario público	0.0 Comentario privado Comentario público	5.0 (2.44)

Ilustración 25: Vista de calificaciones Profesor

Has iniciado sesión como: **J Manuel Maldonado**

Cerrar sesión

CALIFICACIONES

matematicas

Alumno	trabajo 1 15.0%	Totales 15.0%
J Manuel Maldonado	7.5 Fallaste en el punto 2	1.125 (7.5)

Ilustración 26: Vista de calificaciones Alumno

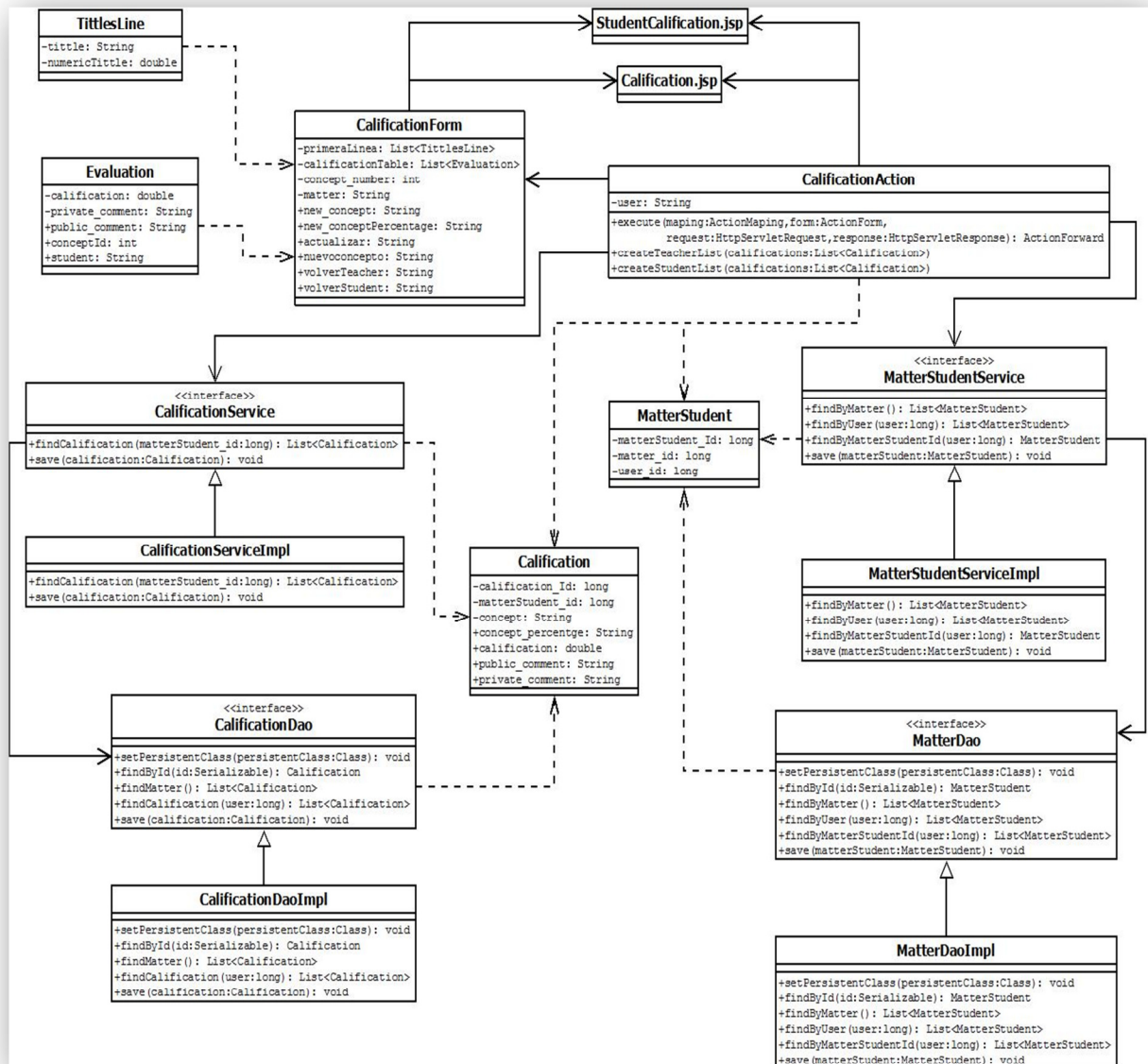


Ilustración 27: UML Calificaciones

La pantalla de calificaciones se muestra con dos interfaces en función de si el usuario es propietario o no de la asignatura a mostrar.

- ❖ Si el usuario es propietario: Significa que imparte la asignatura y es responsable de evaluar a los alumnos. El profesor visualizará pues una lista de los alumnos suscritos en la asignatura junto con sus calificaciones en los distintos conceptos. Para añadir un concepto no debe más que introducir su nombre y su porcentaje en el peso de la asignatura para generar una nueva columna. Una vez generada, pueden editarse las calificaciones de los alumnos en cualquier momento, pulsando en el botón Guardar Evaluaciones para que la información persista. También se da la posibilidad de rellenar mediante un tooltip un comentario público (que verá el alumno) y otro privado, solo visualizable por el profesor. Se incluye al final una columna de totales que realiza una suma ponderada de la nota según el porcentaje de cada concepto y el mismo resultado obtenido sobre 10 para una mejor comprensión por parte del alumno.
- ❖ Si el usuario no es propietario: Significa que es alumno de la asignatura y como tal, podrá consultar sus calificaciones en los distintos conceptos que haya previsto el profesor y la suma de su nota ponderada sobre el total y sobre 10. Asimismo, se mostrará el comentario público que haya añadido el profesor sobre la calificación si es que lo hubiera.

Esta pantalla ofrece la posibilidad de volver a la página de lecciones mediante el botón “Ver lecciones”. Si eres profesor, también se mostrarán los botones “Ver docencia”, que te redireccionará a la página de docencia y el botón “Guardar Evaluaciones” para salvar los cambios efectuados.

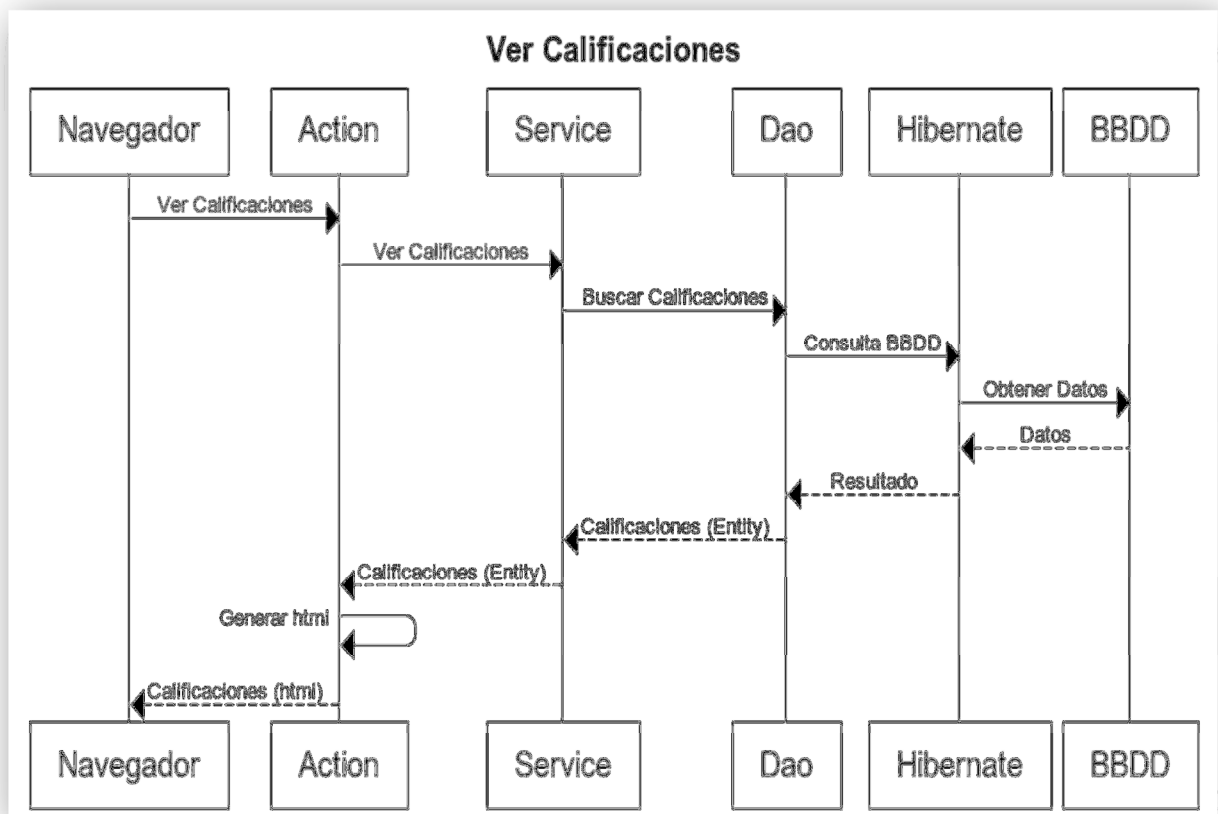


Ilustración 28: Diagrama de secuencias ver calificaciones

Desde el navegador, el usuario invoca la acción de ver las calificaciones. Esta petición es recogida por el Action, que invocará a los servicios de la aplicación. Estos servicios traducirán la petición para que pueda ser entendida por los Dao, que realizarán la llamada a Hibernate para que obtenga los datos de la base de datos. El resultado se transformará al modelo de datos de la aplicación, almacenando la respuesta en una entity, que será enviada al service y a su vez al action. El action convertirá la Entity al modelo de datos de la Vista, especificado en los Forms y generará el código html a través de la JSP que se mostrará en el navegador.

4.7.5. Vista Cronograma

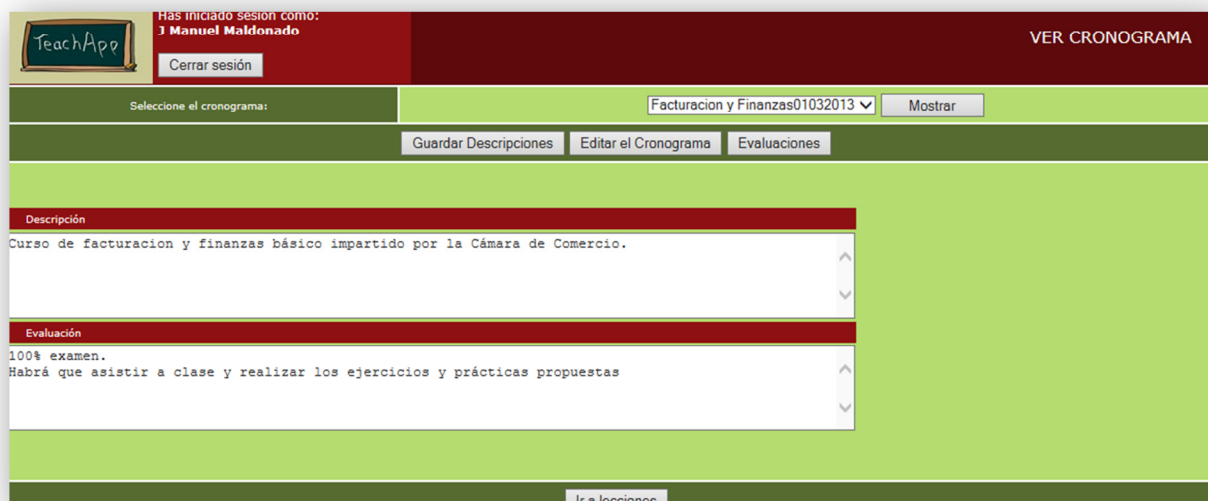


Ilustración 29: Vista cronograma

Esta será la pantalla de referencia del profesor. En ella se mostrará una lista de las asignaturas para las cuales es profesor, pudiendo visualizar las descripciones y la evaluación de cada una de ellas pulsando el botón 'mostrar'. Si lo desea, puede editar estas descripciones y guardar los cambios pulsando 'Guardar Descripciones'.

Además, se mostrará un botón 'Editar el Cronograma', que al pulsarlo redirigirá a la página de edición de cronogramas, para modificar el cronograma de la asignatura seleccionada.

Si se desea, se puede crear un nuevo cronograma seleccionando la opción 'nuevo cronograma' del desplegable de asignaturas. Esto redireccionará a la página de creación de cronogramas.

También es posible acceder a las calificaciones de los alumnos suscritos a cada asignatura pulsando en el botón 'calificar'; este botón redireccionará a la página de calificaciones.

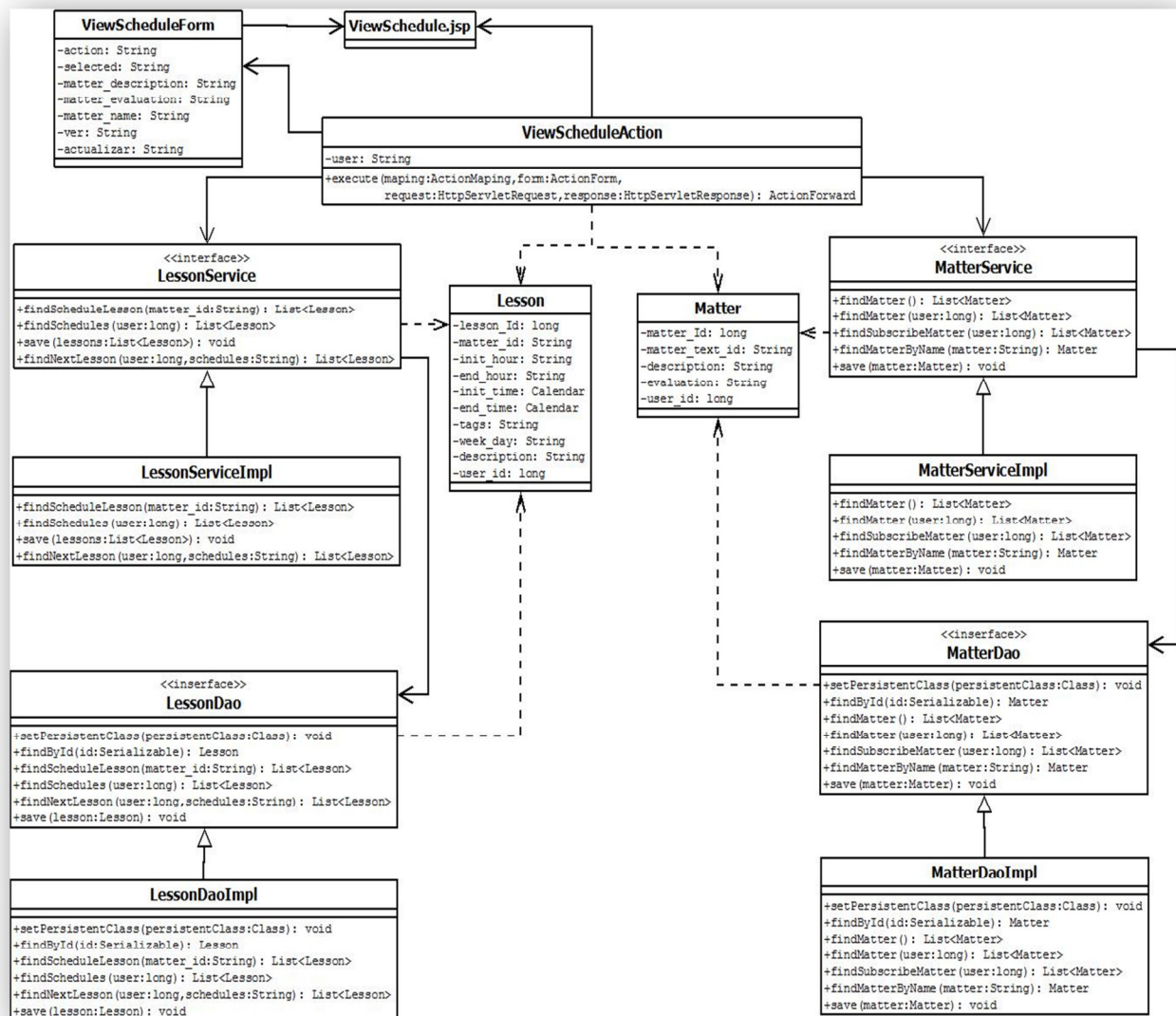


Ilustración 30: UML vista cronograma

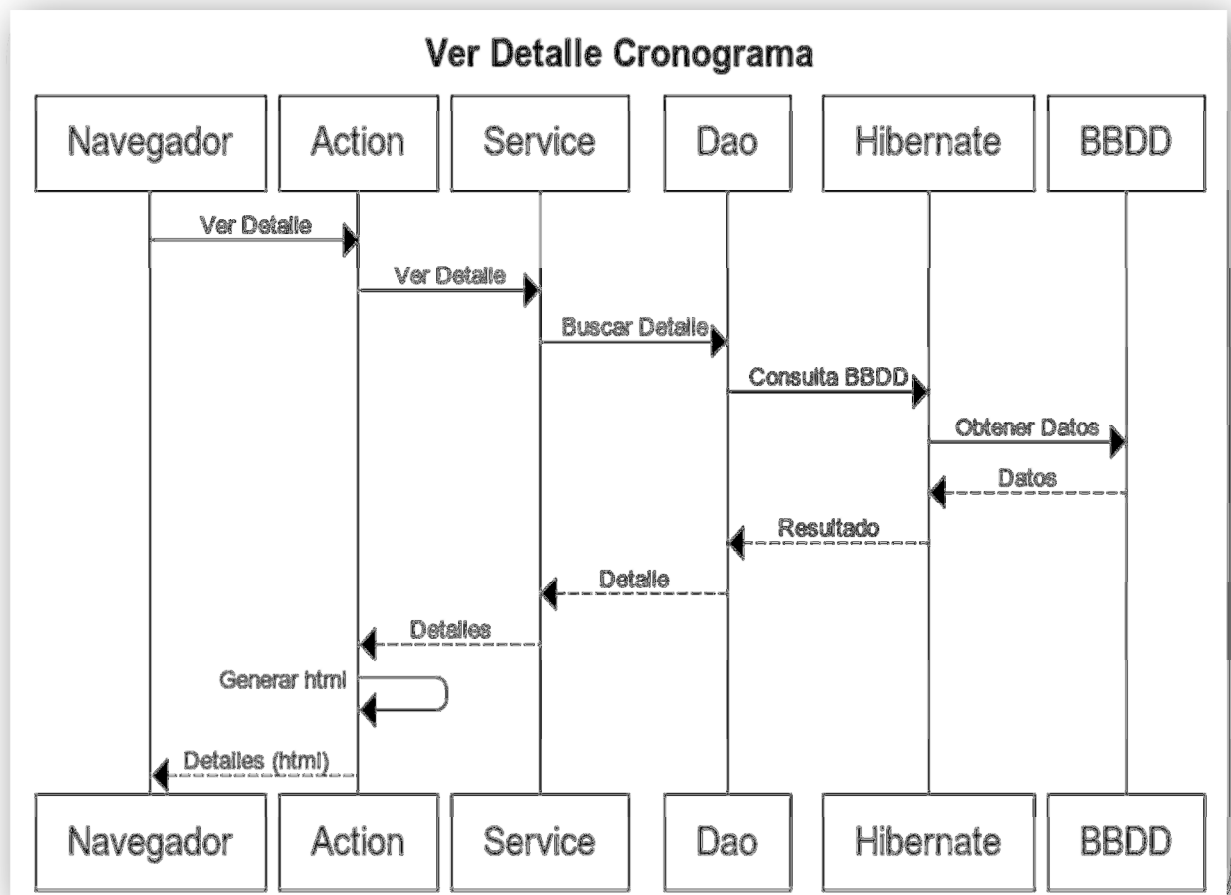


Ilustración 31: Diagrama de secuencia detalle cronograma

Para ver el detalle de un cronograma habrá que seleccionar en el menú desplegable del navegador la asignatura de la cual queremos obtener la información y pulsar sobre el botón mostrar. Una vez hecho esto, el formulario se resolverá en el action, el cual pasará la petición al servicio indicando la asignatura a mostrar. El servicio enviará una petición al dao para que recupere de base de datos a través de Hibernate los detalles de la asignatura facilitada. Dichos detalles retornarán a través de las distintas capas, traducándose a textos en html a mostrar por el navegador.

4.7.6. Nuevo Cronograma

Has iniciado sesión como:
J Manuel Maldonado
 Cerrar sesión

CREAR UN NUEVO CRONOGRAMA

Introduza el nombre del curso

Curso de Arte

Introduce la fecha de inicio del curso:

01/09/2014

¿Cuántas semanas dura el curso?

10

Día	Hora de Inicio	Hora de Fin	Descripción
Lunes			
Martes	16:00	17:00	
Miercoles			
Jueves			
Viernes	15:00	17:00	
Sabado			
Domingo			

Generar Cronograma Preliminar

Ilustración 32: Vista nuevo cronograma

En esta pantalla se definirá las características principales del cronograma para la creación de una nueva asignatura ofertada en la aplicación. Por ello, se debe rellenar el nombre del curso, así como su fecha de inicio y las semanas que durará. Opcionalmente se puede rellenar en esta ventana el horario semanal en el que se impartirán las clases, así como una descripción general de las mismas, aunque esto puede ser modificado más adelante. Al pulsar sobre Generar Cronograma Preliminar, se avanzará hasta la pantalla de “Editar el Cronograma” que explicaremos a continuación.

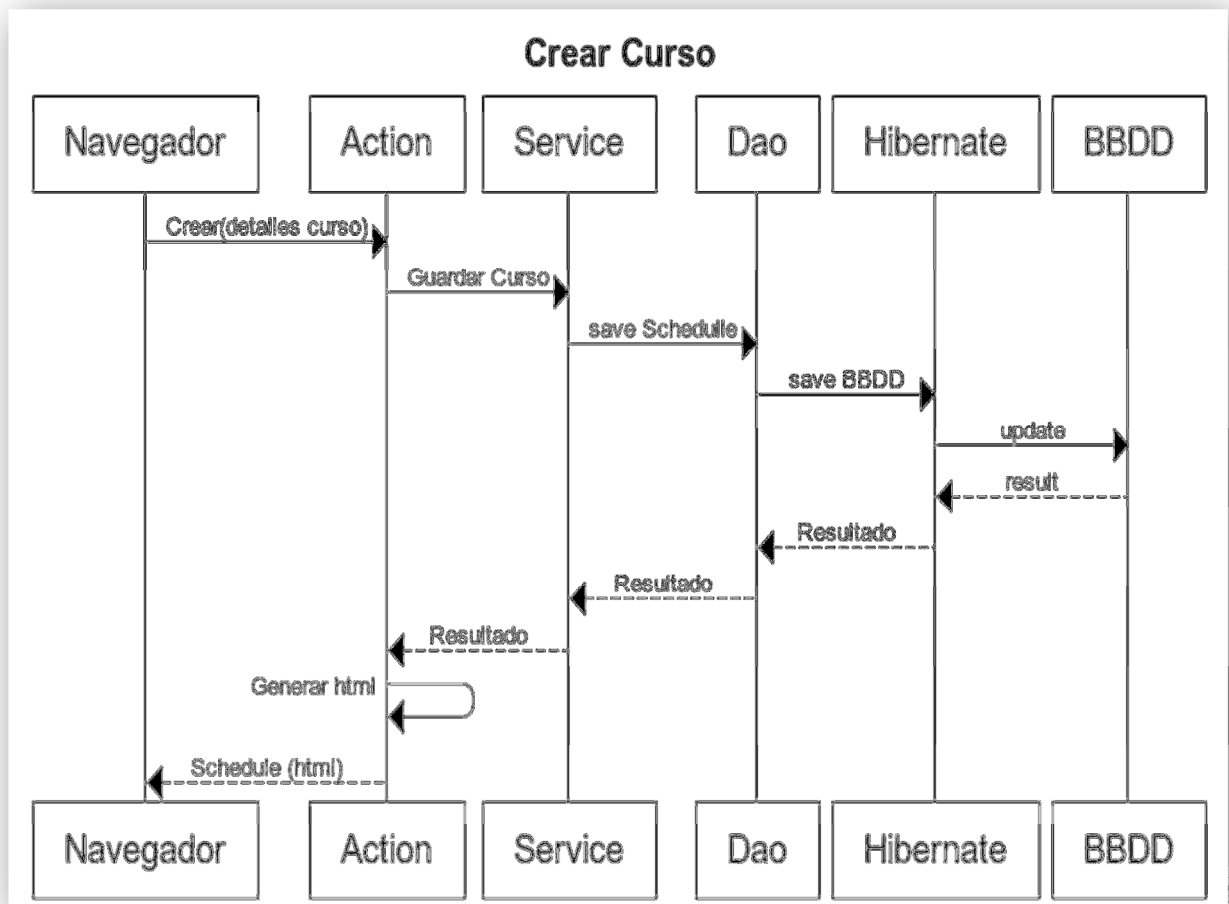


Ilustración 34: Diagrama de secuencia crear curso

Desde el navegador se rellenarán los campos obligatorios del formulario de creación de cursos y se pulsará sobre el botón "generar cronograma preliminar". Una vez hecho esto, se llamará al action que realizará la petición de creación de curso al servicio, que será el encargado de recopilar toda la información necesaria para registrar el curso y sus lecciones. Una vez completada esta fase, se enviará la información a los Daos, que se encargarán a través de Hibernate de registrarlo todo en base de datos para obtener persistencia. El resultado se devolverá a través de las distintas capas de la aplicación y se generará el html necesario para que el navegador pueda mostrar la información procesada del curso generado.

4.7.7. Editar Cronograma

Has iniciado sesión como: **J Manuel Maldonado**

[Cerrar sesión](#)

EDITAR EL CRONOGRAMA: Facturacion y Finanzas01032013

Subir un Recurso **Lección:** Introduccion **Archivo:** [Examinar...](#) [Subir fichero](#)

Semana 1				
Día	Hora de Inicio	Hora de Fin	Descripción	Recursos
Lunes				Ver
Martes	13:00	14:00	Introduccion	Ver
Miercoles				Ver
Jueves	18:00	20:00	Historia	Ver
Viernes				Ver
Sabado				Ver
Domingo				Ver

Semana 2				
Día	Hora de Inicio	Hora de Fin	Descripción	Recursos
Lunes				Ver
Martes	13:00	14:00	Conceptos basicos de facturacion	Ver
Miercoles				Ver
Jueves	18:00	20:00	aspectos legales de facturacion	Ver
Viernes				Ver
Sabado				Ver
Domingo				Ver

Recursos disponibles: [tutoria pfc.txt](#)

Ilustración 35: Vista editar Cronograma superior

Sabado				Ver
Domingo				Ver

Semana 10				
Día	Hora de Inicio	Hora de Fin	Descripción	Recursos
Lunes				Ver
Martes	13:00	14:00	practica con contapulus	Ver
Miercoles				Ver
Jueves	18:00	20:00	examen final	Ver
Viernes				Ver
Sabado				Ver
Domingo				Ver

[Guardar](#) [Exportar a ICS](#)

Ilustración 36: Vista editar Cronograma inferior

La pantalla de editar el cronograma será accedida por primera vez tras haber generado un nuevo cronograma o bien cuando se seleccione la opción editar desde la página de docencia (Vista Cronograma). En ambos casos, se mostrarán todas las semanas en las que se desarrolla el curso así como las horas en las que se imparten las clases y el temario previsto. Aparecerán marcados en rojo de forma predeterminada los fines de semana, aunque pulsando sobre el botón de la derecha podremos marcar cualquier día como festivo. De la misma forma, podemos marcar una semana entera como no lectiva pulsando sobre el tooltip que aparecerá al navegar sobre el número de la semana. Si navegamos sobre la palabra "Ver" en la columna recursos, aparecerán los recursos subidos por el profesor para esa lección. Para subir un recurso habrá que seleccionar en el menú superior la lección a la cual queremos asociar el recurso, así como seleccionar el archivo a compartir con el botón examinar. Al pulsar el botón "Subir fichero" se almacenará el recurso en Dropbox y este estará disponible para ser descargado desde la página "Ver lección" tanto por el alumno como por el profesor.

La pantalla ofrece los botones "Guardar" el cual salvará los cambios. Y el botón "Exportar a ICS", que generará un fichero ICS en formato estándar para añadir al calendario con la información de las lecciones.

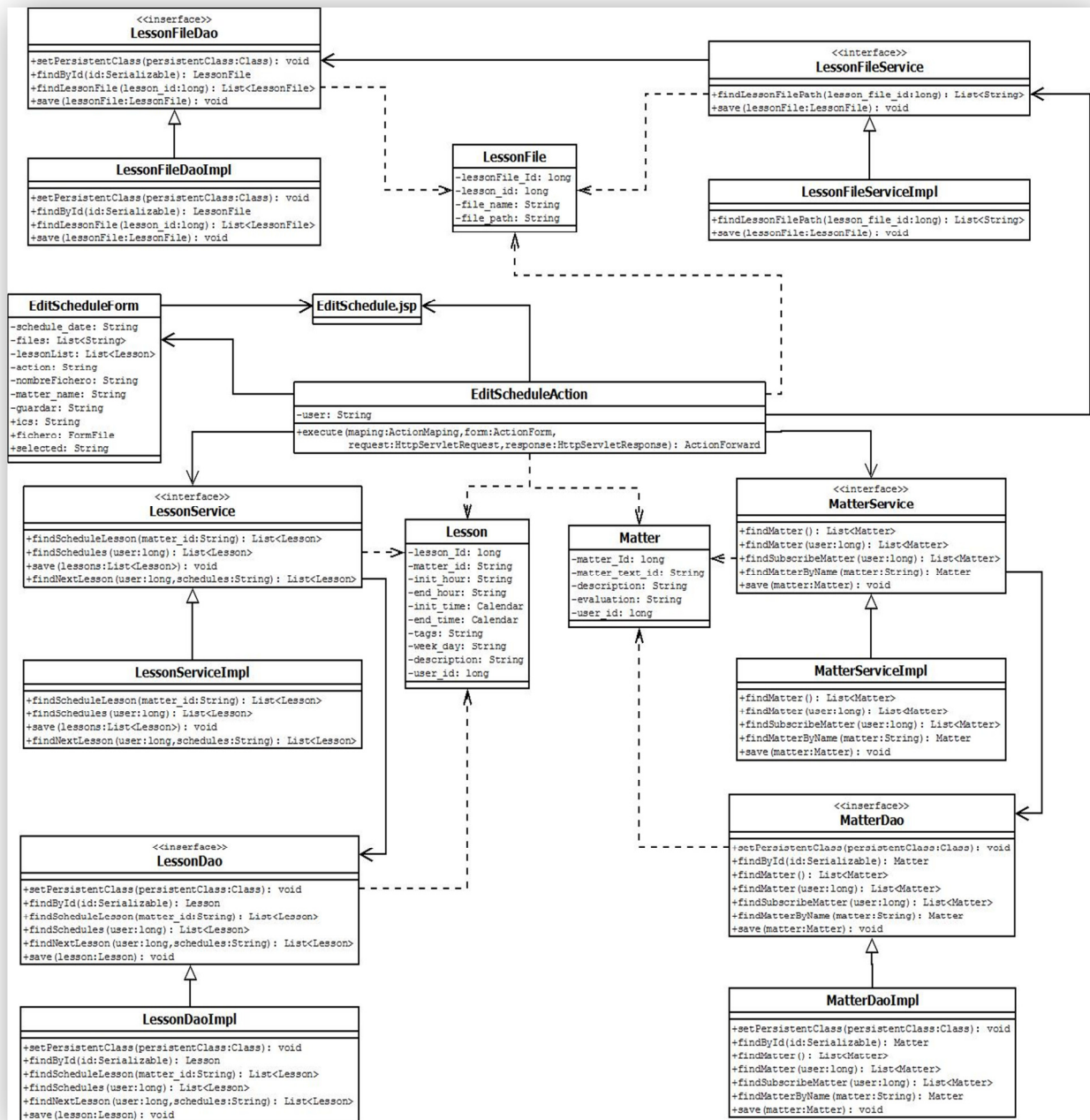


Ilustración 37: UML edición cronograma

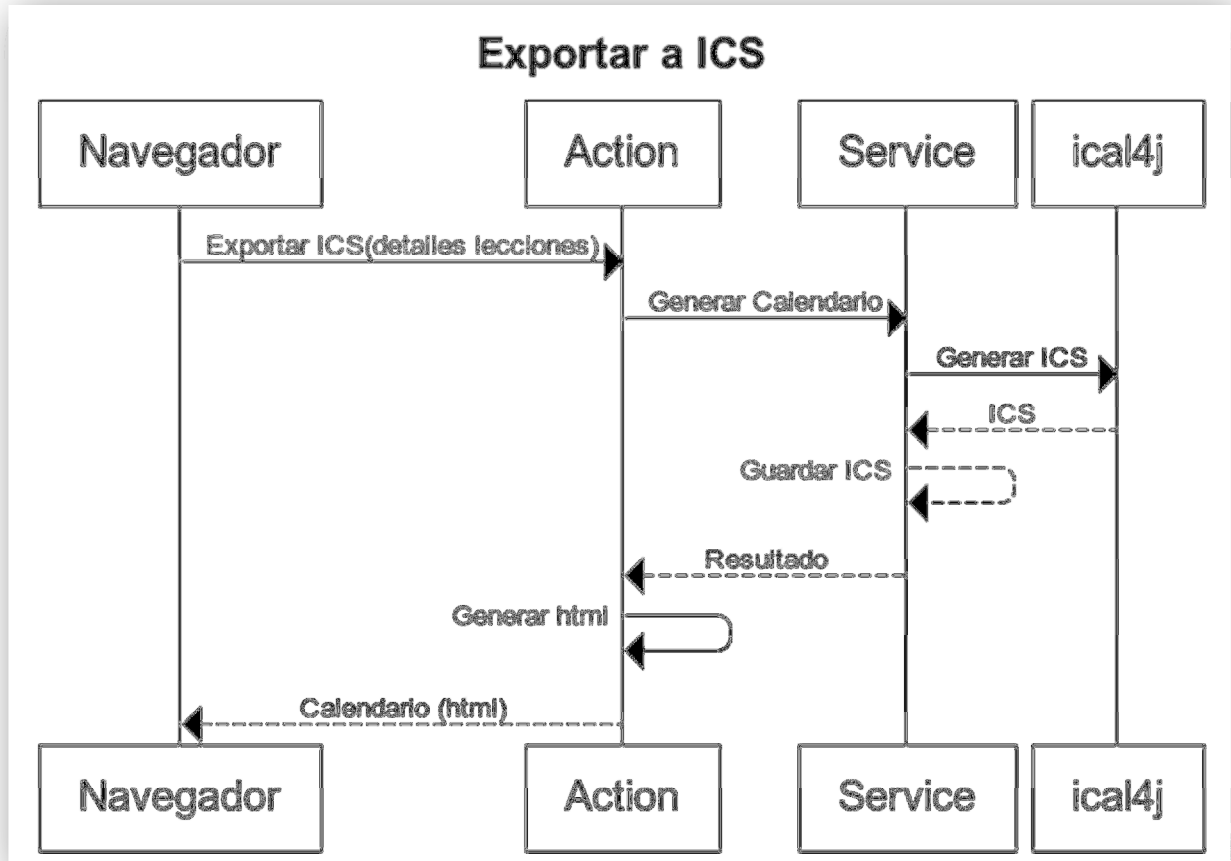


Ilustración 38: Diagrama de secuencia de exportación a ICS

Desde el navegador al pulsar sobre el botón “Exportar ICS” se llama al action que repercute la acción en el servicio. El servicio estructura la información y llama al API de ical4j, que es una librería que se encarga de generar el calendario en formato ICS. Una vez creado el calendario en ICS, el servicio se encarga de almacenarlo antes de devolver la acción al navegador, que mostrará en código html el resultado de la acción.

5. Pruebas

La realización de las pruebas está ligada a la metodología empleada. Para cada tarea se realizarán pruebas de aceptación de forma que se cumpla el DoD pactado por el equipo. Esto y la posterior demo con el tutor, garantizan la validación de las historias de usuario abordadas.

Además de estas pruebas unitarias, en cada Sprint se realizan unas pruebas de aplicación, que contemplarán el paso de una checklist que contenga todas las funcionalidades implementadas hasta el momento. De esta forma nos cercioraremos de que la puesta en producción de nuevos requisitos no devaluará ni perjudicará a la calidad del producto ya entregado. Estas pruebas también servirán para identificar bugs que puedan aparecer mediante el uso de la aplicación.

5.1. Definition of Done

Para poder validar y entregar una tarea se debe definir cuándo se debe dar por concluida la misma y qué criterios debe de seguir para ser aceptada y resuelta. Con este objetivo, se definen una serie de requisitos o validaciones que toda tarea debe de cumplir para considerarse cerrada:

- ❖ Que cumpla con todos los requisitos de la tarea: se validará con el profesor que la funcionalidad entregada es la deseada y especificada.
- ❖ Que funcione correctamente: se validará con distintas pruebas funcionales que el desarrollo responda a distintas acciones ejercidas por parte del usuario incluso cuando no se sigue el flujo esperado.
- ❖ Que no influya en el correcto funcionamiento de otras funcionalidades: en cada entrega se deberá superar una checklist en las que se comprobaran todas las funcionalidades desarrolladas hasta la fecha para la aplicación, detectando en este punto si los nuevos desarrollos influyen negativamente en los ya entregados.
- ❖ Que se mantenga la estructura y la calidad del código: será responsabilidad del programador revisar y refactorizar los elementos necesarios del código de forma que se siga escrupulosamente el diseño de la arquitectura así como todas las buenas prácticas de la programación con objetos de Java.
- ❖ Que la tarea este documentada y su código adecuadamente comentado.

5.2. Procedimiento de validación de una tarea

El procedimiento para validar una tarea es cumplir los puntos definidos anteriormente en el DoD. Para ello se deberán de seguir los siguientes pasos.

Calidad de Código

- ❖ Revisión de la arquitectura y estructuración del código.
- ❖ Revisión de duplicidades.
- ❖ Revisión del maquetado.
- ❖ Revisión de los comentarios.
- ❖ Revisión de estándares de nombrado y buenas prácticas de Java.

Refactorización

- ❖ Revisión de código antiguo y reubicación o modificación del mismo si es necesario.
- ❖ Reutilización de métodos y limpieza de código no necesario.
- ❖ Simplificación y unicidad de funciones.
- ❖ Uso de patrones de diseño y estructuras optimizadoras.

Validación técnica

Para la validación técnica se desplegará la aplicación en un servidor local Tomcat utilizando la base de datos cargada en MySQL. Con la función "deploy" definida usando MAVEN en el pom del proyecto, se podrá desplegar el war para poder visualizar la aplicación en el navegador. Se deberá comprobar que:

- ❖ La aplicación despliega sin errores.
- ❖ La interfaz no muestre desperfectos o anomalías.
- ❖ Que se pueda navegar por las funcionalidades sin provocar errores.
- ❖ Que las funcionalidades realicen acciones y que sean las esperadas.
- ❖ Que haya persistencia de las acciones en su caso.
- ❖ Que el rendimiento sea el adecuado y que no se provoquen problemas con el hardware o con otro software.

Checklist

Para verificar que las funcionalidades ya entregadas y que la nueva tarea se integra adecuadamente en la aplicación, se pasará una checklist consistente en verificar todas las funcionalidades de la aplicación en sus flujos habituales para asegurarnos de que no se ha corrompido nada desde la anterior reléase. Se puede encontrar un anexo con la checklist de la aplicación al final del documento.

Validación funcional

Se validará en presencia del profesor que los requisitos se han entendido y reflejado correctamente en la aplicación, de forma que la página final recoja las funcionalidades y muestre la información necesaria al usuario. Esta validación tendrá en cuenta las entradas y posibles salidas de las funcionalidades implementadas:

RUF-01 Crear un Cronograma del curso	
Entradas	Nombre Fecha de inicio Nº de semanas de duración
Proceso	1. Validación de entradas. 2. Creación del curso <Nombre> con fecha <Fecha de inicio>. 3. Creación de <Nº semanas de duración> *7 lecciones del curso.
Procesos alternativos	1.B. Validación de entradas incorrecta
Salida	Página web con el nombre del curso con todas sus lecciones agrupadas por semana.
Salidas de error	Formulario inicial con mensaje de error en el campo incorrecto.

RUF-02 Modificación de un Cronograma de curso	
Entradas	Descripciones Días de la semana Horas de inicio Horas de fin Ids de lecciones
Proceso	1. Validación de entradas. 2. Por cada lección: Actualización de datos de la lección con <Descripción>, <Día de la semana>, <Hora de inicio> y <Hora de fin>.
Procesos alternativos	1.B. Validación de entradas incorrecta.
Salida	Página web con los datos del cronograma actualizados.
Salidas de error	Formulario inicial con mensaje de error en el campo incorrecto.

RUF-03 Marcar, desmarcar días no lectivos	
Entradas	Ids de lecciones
Proceso	Por cada lección: 1. Si lección lectiva, marcar como no lectiva.
Procesos alternativos	Por cada lección 1.B. Si lección no lectiva, marcar como lectiva.
Salida	Página web con los datos de las lecciones lectivas actualizados.
Salidas de error	No ha sido posible realizar el requerimiento.

RUF-04 Exportar un cronograma a un calendario ICS	
Entradas	Cronograma
Proceso	1. Validación del cronograma. 2. Conversión de lecciones en reuniones ICS. 3. Guardado del calendario en un archivo con extensión ICS.
Procesos alternativos	1.B. Validación del cronograma incorrecta. 3.B. Error al guardar el documento
Salida	Fichero ICS con el cronograma del curso
Salidas de error	1. Formulario inicial con mensaje de error en el campo incorrecto. 3. Error de E/S al crear el fichero.

RUF-05 Modificar información curso	
Entradas	Descripción Evaluación Id del curso
Proceso	1. Validación de entradas. 2. Actualización de datos del <id curso> con <Descripción>, <Evaluación>
Procesos alternativos	1.B. Validación de entradas incorrecta.
Salida	Página web con los datos del curso actualizados.
Salidas de error	Formulario inicial con mensaje de error en el campo incorrecto.

RUF-06 Publicar recursos en lecciones	
Entradas	Id de lección Recurso
Proceso	1. Validación de entradas. 2. Conexión con Dropbox 3. Subida de <Recurso> a Dropbox 4. Registro de recurso en el sistema: Nombre, dirección, <id de lección>
Procesos alternativos	1.B. Validación de entradas incorrecta. 2.B. Conexión con Dropbox incorrecta. 3.B. Interrupción de la comunicación.
Salida	Formulario inicial con el nombre del recurso subido en la lección
Salidas de error	1. Formulario inicial con mensaje de error en el campo incorrecto. 2. No ha sido posible establecer la conexión con el servidor de contenidos 3. Se ha interrumpido la conexión con el servidor de contenidos.

RUF-07 Autenticar usuario con Facebook	
Entradas	Nombre de usuario Contraseña
Proceso	1. Redirección a página de login de Facebook 2. Introducción de parámetros de entrada 3. Validación por parte de Facebook del usuario 4. Redirección a página de TeachApp
Procesos alternativos	1.B. Redirección incorrecta. 3.B. Usuario incorrecto: 3 reintentos.
Salida	Usuario autenticado con su nombre y mail en su sesión
Salidas de error	1. Página no encontrada 3. No se ha podido iniciar sesión

RUF-08 Matricular cursos	
Entradas	Id Cursos seleccionados Id Alumno
Proceso	1. Por cada <id de curso seleccionado> 2. Matricular al <id de Alumno> en el <id curso> 3. Crear conceptos de evaluación pasados.
Procesos alternativos	3.B No hay conceptos de evaluación pasados
Salida	Lista de cursos matriculados incluye los cursos seleccionados. Evaluación para esos cursos incluye "0" en conceptos de evaluación ya pasados.
Salidas de error	No ha sido posible realizar la operación.

RUF-09 Información próxima clase	
Entradas	Ids cursos seleccionados Fecha actual
Proceso	1. Recuperar lecciones de cursos seleccionados 2. Ordenar la lista cronológicamente por fecha del curso 3. Mover el puntero a la primera lección posterior a la <fecha actual>
Procesos alternativos	1.B. Si no hay cursos seleccionados: entonces, seleccionar todos. 1.C. Si así tampoco hay cursos, redirección a la página de matriculación.
Salida	Información detallada de la lección: día, hora, información del curso, temario, recursos, etc.
Salidas de error	No hay próximas lecciones

RUF-10 Resumen próximas lecciones	
Entradas	Ids cursos seleccionados Fecha actual
Proceso	1. Recuperar lecciones de cursos seleccionados 2. Ordenar la lista cronológicamente por fecha del curso 3. Mover el puntero a la segunda lección posterior a la <fecha actual>
Procesos alternativos	1.B. Si no hay cursos seleccionados: entonces, seleccionar todos. 1.C. Si así tampoco hay cursos, redirección a la página de matriculación. 3.B. Si no hay próximas lecciones, volver al inicio.
Salida	Lista de las siguientes 5 lecciones con información de: fecha, hora, curso y temario.
Salidas de error	No hay próximas lecciones

RUF-11 Descargar recursos	
Entradas	Id recurso
Proceso	1. Recuperación del registro <id recurso>. 2. Conexión con Dropbox. 3. Descarga del recurso de la ruta del registro en Dropbox.
Procesos alternativos	2.B. Conexión con Dropbox incorrecta. 3.B. Interrupción de la comunicación.
Salida	Ventana emergente preguntando si abrir o guardar el recurso.
Salidas de error	2. No ha sido posible establecer la conexión con el servidor de contenidos 3. Se ha interrumpido la conexión con el servidor de contenidos.

RUF-12 Gestionar Evaluaciones alumnos	
Entradas	Concepto % de peso Calificación Comentarios privados Comentarios públicos Id Alumnos
Proceso	1. Validación de datos. Por cada alumno de alumno: 2. Registrar el <concepto>, el <% de peso>, la <calificación>, el <comentario privado> y el <comentario publico>
Procesos alternativos	2.B. No hay alumnos matriculados.
Salida	Lista de alumnos con sus calificaciones y comentarios ordenados por concepto.
Salidas de error	Formulario inicial con mensaje de error en el campo incorrecto.

RUF-13 Visualizar evaluaciones	
Entradas	Id curso Id alumno
Proceso	1. Búsqueda de conceptos para el <id curso> 2. Recuperación de calificaciones para <id alumno> de los conceptos.
Procesos alternativos	1.B No hay conceptos.
Salida	Calificaciones y comentarios públicos ordenados por concepto.
Salidas de error	No hay conceptos evaluados para este curso.

6. Conclusiones

En las conclusiones se exponen las dificultades acaecidas durante el proyecto y el grado de consecución de objetivos del mismo. Se recopilan las experiencias derivadas del método de trabajo empleado y del uso de las tecnologías definidas. Se hace balance del trabajo de investigación realizado y de las aportaciones del proyecto al sector del E-Learning; asimismo, se exponen los puntos de evolución futura en el área y las carencias de los actuales servicios. Por último se plantea una opinión personal al respecto en la que se enumeran las aportaciones del proyecto a la evolución académica y personal del alumno.

6.1. Dificultades

Durante el desarrollo del trabajo se han encontrado diversas dificultades a la hora de diseñar e integrar las tecnologías en el sistema. Algunas de las cuales se detallan a continuación:

- ❖ Alcance del proyecto. Puesto que la amplitud del proyecto es demasiado grande como para ser abordada por una única persona en el tiempo que dura un proyecto de fin de carrera, ha sido necesario limitarlo para responder a las necesidades más elementales, quedándose en el tintero una gran cantidad de posibilidades e ideas de mejora.
- ❖ Falta de documentación. Puesto que algunas de las tecnologías empleadas son muy recientes, aún no es posible encontrar documentación clara y suficiente para resolver las dudas de implementación que generaban. Así, hemos tenido problemas con cambios de versiones de APIs, por ejemplo con Dropbox. Y con la integración de redes sociales en entornos no productivos, tuvimos que utilizar Facebook cuando la idea inicial era usar Twitter para el inicio de sesión.
- ❖ Montaje de plataforma. Se han tenido algunos problemas a la hora de montar los servidores (de aplicación y de base de datos) y encontrar los plugins necesarios para correrlos y enlazarlos con el resto de tecnologías. Esto ha sido así dado que se ha usado un equipo con Windows 7 funcionando en 64 bits y es difícil encontrar soporte en algunos aspectos para esta arquitectura.
- ❖ Compatibilidad en navegadores. Hemos sufrido algunas demoras al tratar de hacer compatibles algunos elementos con Internet Explorer. Además el cambio de versión a IE10 publicada durante la realización del proyecto hizo necesaria la refactorización de algunas páginas para que fueran compatibles.
- ❖ Diseño e integración de funcionalidades. Algunas de las páginas de la aplicación integran una gran variedad de funcionalidades que hemos debido de integrar de forma que el usuario pueda emplearlas de forma sencilla. Esto ha propiciado el rediseño de páginas enteras según se añadía funcionalidad para que siguieran siendo intuitivas y manejables.



Conciliación de horarios. Debido a que debo conciliar la realización del proyecto con mi actividad laboral, me ha resultado difícil la comunicación directa con el tutor en la universidad, así como la dedicación del tiempo necesario para avanzar con el proyecto en momento determinados.

6.2. Consecución de objetivos

Al inicio de la memoria se plantearon tres grupos de objetivos, que se pasarán a comentar a continuación:

Desarrollo de una plataforma Web que facilite la gestión

A grandes rasgos, se ha conseguido crear una plataforma web fácilmente utilizable que ofrece una gran cantidad de posibilidades y funcionalidades que pueden ser muy útiles en todas las labores de gestión del profesor. Se ha conseguido una herramienta sencilla de creación de cronogramas de asignaturas que pueden ser exportados a calendarios para la comodidad del docente. Sobre ese cronograma pueden añadirse recursos que serán visualizados por los alumnos. Además puede evaluarse de forma sencilla a los alumnos, registrando las calificaciones que han obtenido en los diversos ejercicios y poniéndolas en su conocimiento de forma inmediata.

Apoyo de herramientas tecnológicas

Se han integrado gran variedad de tecnologías, tanto a nivel de plataforma, con la utilización de un servidor Tomcat con base de datos en MySQL; como a nivel de desarrollo, con la utilización de un entorno de programación J2EE con el apoyo de herramientas como Struts, Spring, Maven, Hibernate, etc.; como a nivel funcional, con la integración de Facebook, Dropbox y la posibilidad de incluir contenido de Google Drive, Youtube, Hangouts y otros elementos interactivos.

Con todos estos elementos mencionados se ha conseguido un nivel tecnológico alto que se mantendrá puntero durante bastante tiempo, dado que se han utilizado las versiones más modernas de estas tecnologías.

Independencia de la plataforma tecnológica

Puesto que este proyecto aporta una plataforma propia, fácilmente montable en cualquier servidor debido a la sencillez de configuración de la misma, se ha independizado la aplicación de otras plataformas existentes en universidades u otros centros docentes. Sin embargo, dada la arquitectura empleada y su disposición web, sería fácilmente integrable en dichos sistemas. Del mismo modo, puesto que internet es una red universal, puede ser accedida desde cualquier dispositivo, arquitectura y sistema operativo que cuenten con un navegador web estándar.

6.3. Aportaciones al E-Learning

Como vimos en el estado del arte, hay cuatro puntos primordiales sobre los que nuestro proyecto se ha enfocado en el E-Learning para aportar algo nuevo y con valor añadido. Vamos a repasar que ofrece este proyecto para solventar los problemas detectados.

- ❖ **Independencia:** La plataforma no depende de ninguna entidad educativa y puede instalarse fácilmente en cualquier servidor para ser accedida vía Internet por cualquier usuario.
- ❖ **Universalidad:** Al ser una aplicación web, puede ser manejada desde cualquier navegador, independientemente de la arquitectura y del sistema operativo que se utilice. Esto permite que pueda ser accedida desde ordenadores personales, tablets, smartphones, etc.
- ❖ **Versatilidad:** Se han integrado numerosas soluciones tecnológicas y redes sociales. Así, el proyecto integra directamente Facebook y Dropbox, además de dar soporte a otras tecnologías como Youtube, Hangouts, Google Drive, etc.
- ❖ **Estructuración:** La creación de cronogramas exportables a los calendarios estándar, la organización en lecciones y las herramientas de matriculación y evaluación de la aplicación responden a todas las necesidades estructurales que necesita una buena plataforma de E-Learning.

6.4. Evolución futura

Como hemos mencionado, el ámbito del proyecto es relativamente pequeño en comparación con las posibilidades que se ofrecen, dado que era imposible abarcar todo en un proyecto de fin de carrera. Así, queda la posibilidad de complementar y ampliar los horizontes del proyecto en varios campos que expongo a continuación:

- ❖ **Comunicaciones:** Pueden ampliarse las características en esta área incluyendo un buzón de comunicaciones privadas, foros, chats y cualquier otro elemento que permita una comunicación síncrona y asíncrona más efectiva en la aplicación.
- ❖ **Red Social:** Podría potenciarse el aspecto de red social de la plataforma implementando perfiles y motivando a los usuarios competitivamente por niveles de aprendizaje, veteranía o calidad de sus cursos.
- ❖ **Integración tecnológica:** pueden integrarse más redes sociales tanto para el inicio de sesión como para compartir hitos y eventos con los amigos, por ejemplo Twitter. Asimismo es necesario no perder el foco en la integración de las últimas herramientas tecnológicas que vayan apareciendo en el mercado, por ejemplo, nuevas versiones en las API de Dropbox y de Facebook.
- ❖ **Mejora de la interfaz gráfica y adaptación específica para resoluciones pequeñas.** Desarrollando un diseño no basado en parrilla y procurando que se adapte mediante hojas de estilos a distintas resoluciones de pantalla y navegadores.

Todas estas características pueden ser añadidas de forma sencilla en la aplicación dado que se ha usado una arquitectura que favorece el desarrollo y el mantenimiento futuro, por lo que la refactorización y migración a nuevas tecnologías no debería ser especialmente complicado de abordar.

6.5. Conclusiones generales

Creo que se han cumplido los objetivos iniciales del proyecto y que incluso se han complementado enriqueciéndose a medida que iban tomando forma. A esta evolución ha contribuido la metodología ágil empleada, que organizaba en iteraciones el proyecto y propiciaba la retroalimentación y versatilidad de prioridades, descubriéndose muchas posibilidades y tecnologías para dar una vuelta de tuerca a las capacidades implementadas. El resultado de este proyecto de fin de carrera es una plataforma de E-Learning útil, innovadora y que incorpora a nivel de desarrollo modernas tecnologías y buenas prácticas que permiten una evolución futura y un mantenimiento simple de abordar.

A nivel educativo este proyecto me ha servido para gestionar de principio a fin las complejidades de un gran trabajo, desde la fase de análisis de necesidades, hasta la entrega final del mismo. Asimismo, he podido conocer y profundizar en diversas tecnologías, así como montar e implantar una plataforma adecuada, que es algo que raramente tenemos la posibilidad de efectuar. La experiencia adquirida en los problemas enfrentados me servirá en el mundo laboral para ser más competente y dar una respuesta eficaz a los problemas que se presenten.

A nivel personal el desempeño del proyecto ha supuesto el fin de mi etapa académica con la consecución del título Universitario, que abrirá nuevas puertas a mi carrera laboral que ya desempeño. La interacción con personas ampliamente capacitadas como es el tutor de mi proyecto me ha permitido aprender a trabajar con otros puntos de vista y a conocer las necesidades del cliente al desarrollar su producto.

En general, me ha entusiasmado la realización del proyecto, al ser el E-Learning un tema que me apasiona y sobre el cual me siento muy agradecido de haber podido aportar mi propia visión y mi granito de arena.

7. Referencias

7.1. E-Learning

- ◆ Anderson T., Elloumi, F. (Ed.) (2004). Theory and Practice of Online Learning. Edmonton: Athabasca University.
- ◆ Bates, M. (1989). The design of browsing and berrypicking techniques for the online search interface. Online Review 13, num 5.
- ◆ Berman, P. (2006). E-learning Concepts and Techniques. Bloomsburg, Pennsylvania: Institute for Interactive Technologies.
- ◆ Castañeda, L. y Adell, J. (Eds.) (2013). Entornos personales de aprendizaje: claves para el ecosistema educativo en red. Alcoy: Marfil.
- ◆ Cerner, M. (Ed.) (2011). Learning perspectives 2011. Saratoga Springs (NY): The MASIE Center.
- ◆ Cobb, J. (2010). E-learning 2.0 for associations. Carborro, North Carolina: Tagoras, Inc.
- ◆ Embi, M. A. (2011). Web 2.0 Tools in Education: A Quick Guide. Bangi, Malaysia: University Kebangsaan.
- ◆ Gvirtz, S. y Necuzzi, C. (2011). Educación y tecnologías. Las voces de los expertos. Buenos Aires: Anses.
- ◆ Lowenthal, P. et al. (Ed.) (2009). The CU Online. Denver, Colorado: The University of Colorado Denver.
- ◆ Martínez, J. et al. (2006). Prácticas de e-learning. Barcelona: Octaedro.
- ◆ Masie, E. (Ed.) (2004). 701 e-learning tips. Saratoga Springs (NY): The MASIE Center.
- ◆ Mentzas, G. et al. (2001) "Knowledge networking: a holistic solution for leveraging corporate knowledge", Journal of Knowledge Management, Vol. 5 Iss: 1, pp.94 – 107.
- ◆ Naidu, S. (2003). E-learning: A Guidebook of Principles, Procedures and Practices. Nueva Delhi, India: Commonwealth Educational Media Center for Asia.
- ◆ Oblinger, D. (Ed.) (2005). Educating the Net Generation. Louisville, Colorado: Educause.
- ◆ Oblinger, D. (Ed.) (2012). Game changers: Education and Information Technologies. Louisville, Colorado: Educause.
- ◆ Oblinger, D. (Ed.) (2006). Learning Spaces. Louisville, Colorado: Educause.
- ◆ Paine, N. (Ed.) (2010). Learning Perspectives. Saratoga Springs (NY): The MASIE Center.
- ◆ Paine, N. (2011). Learning Strategies. Saratoga Springs (NY): The MASIE Center.
- ◆ Samaniego, P. et al. (2012). Informe sobre el uso de las TIC en la Educación para Personas con Discapacidad. Quito, Ecuador: UNESCO.

- ◆ Santos Pascualena, M.L. (2006). Organización y gestión de equipos para el desarrollo de contenidos educativos multimedia. Red Digital nº 6.
- ◆ Smith Nash, S. (2009). E-learning Survival Guide. Norman, Oklahoma: Texture Press.
- ◆ Stephenson, J. y Sangrà, A. (2003). Modelos pedagógicos y e-learning. Barcelona: Universitat Oberta de Catalunya.
- ◆ Utrera González, F. (2012). Estrategias Web 2.0 para la enseñanza. Cancún, México: Autoedición.
- ◆ Veletsianos, G. (Ed.) (2010). Emerging Technologies in Distance Education. Edmonton: Athabasca University.

7.2. Tecnologías

- ◆ DropBox: <https://www.dropbox.com/>
- ◆ Facebook: <https://www.facebook.com/>
- ◆ Hibernate: <http://www.hibernate.org/>
- ◆ Maven: <http://maven.apache.org/>
- ◆ MySQL: <http://www.mysql.com/>
- ◆ PlanBox: <https://www.planbox.com>
- ◆ Spring : <http://www.springsource.org/>
- ◆ Struts: <http://struts.apache.org/>
- ◆ Tomcat: <http://tomcat.apache.org/>

7.3. Metodologías

- ◆ Beck, K. (2004). User Stories Applied: For Agile Software Development.
- ◆ Cockburn, A. (2004). Crystal Clear: A Human-Powered Methodology for Small Teams.
- ◆ Cohn, M. (2004). User Stories Applied: For Agile Software Development.
- ◆ Cohn, M. (2009). Succeeding with Agile: Software Development Using Scrum.
- ◆ Fowler, M (1999). Refactoring: Improving the Design of Existing Code.
- ◆ Kniberg, H. (2007). Scrum and XP from the Trenches.
- ◆ Kniberg, H. y Skarin, M. (2010). Kanban and Scrum – Making the Most of Both.
- ◆ Martin, R. (2002). Agile Software Development, Principles, Patterns, and Practices.
- ◆ Martin, R. (2008). Clean Code: A Handbook of Agile Software Craftsmanship.
- ◆ Poppendieck, M. y Poppendieck, T. (2003). Lean Software Development: An Agile Toolkit.

7.4. Estado del arte

- ◆ Aula Global: <https://aulaglobal.uc3m.es/>
- ◆ Joomla: <http://www.joomla.org/>
- ◆ iDoceo: <http://www.idoceo.es/index.php/es/>
- ◆ Duolingo: <http://www.duolingo.com/>
- ◆ Hangout: <http://www.google.com/+/learnmore/hangouts/>

8. ANEXO I: Checklist funcional

Funcionalidad	Funcionamiento	Comentario
LOGIN		
Iniciar Sesión con Facebook	OK	
Redirección a la página correcta	OK	
Cerrar sesión	OK	
VISTA LECCION		
Seleccionar asignaturas a mostrar	OK	
Navegar entre las siguientes lecciones	OK	
Visualizar información de la asignatura	OK	
Ver recursos publicados	OK	
Ver recursos tecnológicos compartidos	OK	
Botón Matricular	OK	
Botón Calificaciones	OK	
MATRICULACION DE CURSOS		
Visualizar asignaturas ofertadas	OK	
Matricular asignaturas	OK	
Botón ver lección	OK	
Botón crear curso	OK	
CALIFICACIONES ALUMNO		
Visualizar fila con conceptos y notas	OK	
Visualizar comentarios públicos de profesor	OK	
Visualizar cálculo de totales	OK	

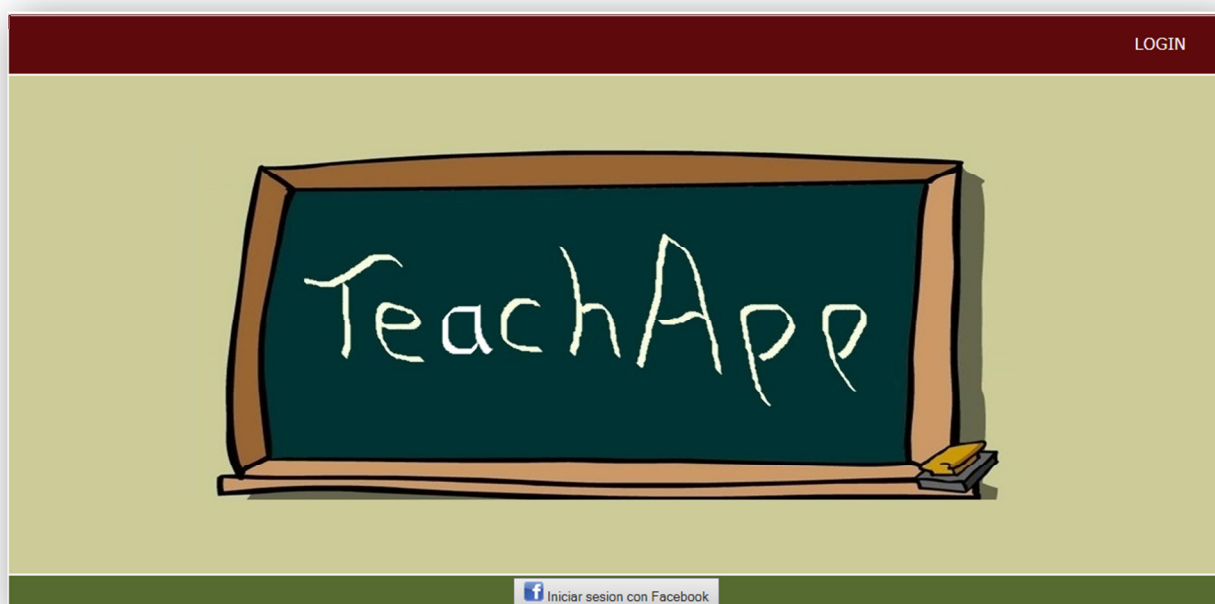
Botón ver lección	OK	
CALIFICACIONES PROFESOR		
Visualizar alumnos matriculados	OK	
Añadir nuevo concepto y porcentaje	OK	
Visualizar y editar conceptos y notas	OK	
Visualizar y editar comentarios públicos de profesor	OK	
Visualizar y editar comentarios privados de profesor	OK	
Visualizar cálculo de totales	OK	
Botón Ver Lección	OK	
Botón Ver Docencia	OK	
Botón guardar cambios	OK	
VISTA CRONOGRAMA		
Visualizar asignaturas impartidas	OK	
Visualizar y editar detalle de asignaturas	OK	
Botón nuevo cronograma	OK	
Botón editar cronograma	OK	
Botón guardar cambios	OK	
NUEVO CRONOGRAMA		
Introducción y validación de campos	OK	
Introducción y validación de patrón semanal	OK	
Botón generar cronograma preliminar	OK	

EDITAR CRONOGRAMA		
Edición y validación de campos de las lecciones	OK	
Marcar/desmarcas día festivo	OK	
Marcar/Desmarcar semana vacacional	OK	
Guardar cambios	OK	
Exportar a calendario	OK	
Subir un nuevo recurso a DropBox	OK	
Visualizar recursos subidos por lección	OK	

9. ANEXO II: Manual del usuario

El presente manual tiene como objeto iniciar al usuario en el manejo de la plataforma de E-Learning Teachapp, de forma que pueda desenvolverse entre las distintas pantallas y funcionalidades que ofrece la aplicación, para poder sacarle el máximo partido y explotar todas sus características.

9.1. Login



Al acceder a la aplicación usted se encontrará con una página similar a esta. Para poder utilizar las funcionalidades de la aplicación, usted debe identificarse como usuario válido de la misma. Para ello, pulse sobre el botón “Iniciar sesión con Facebook”.

IMPORTANTE: Para continuar, usted debe de disponer de una cuenta activa en Facebook, si no es así, cree un usuario en Facebook antes de continuar.

The image shows the Facebook login interface. At the top, there is a blue header with the Facebook logo and a green 'Regístrate' button. Below this, the main content area is white. In the center, there is a box titled 'Entrar en Facebook'. Inside this box, there are two input fields: 'Correo electrónico o teléfono:' and 'Contraseña:'. Below the password field, there is a checkbox labeled 'No cerrar sesión'. To the right of the password field, there is a blue 'Entrar' button and a link 'o Regístrate en Facebook'. Below the 'Entrar' button, there is a link '¿Has olvidado tu contraseña?'. At the bottom of the page, there is a row of language links: Español (España), Català, Euskara, Galego, English (US), Español, Português (Brasil), Français (France), Deutsch, Italiano.

Una vez pulsado el botón anteriormente descrito, el navegador le redireccionará a una página de Facebook segura (conexión https) en la que usted deberá introducir sus datos de acceso a Facebook, esto es: su correo electrónico y su contraseña. Una vez introducidos, pulse en "Entrar". Si los datos son correctos, será redireccionado a la aplicación de TeachApp y estará validado como usuario, pudiendo acceder a las funcionalidades y contenidos de la plataforma. Si los datos no son correctos, podrá volver a introducir sus credenciales de acceso.

9.2. Logout



Una vez identificado como usuario de la aplicación, podrá ver en todo momento su nombre en la esquina superior izquierda de la ventana. Para abandonar la aplicación, es recomendable cerrar la sesión previamente, para ello, pulse en el botón "Cerrar sesión" que se encuentra junto al logo de la aplicación. Será redireccionado a la página inicial.

9.3. Matriculación de Cursos



Una vez iniciada la sesión, si es la primera vez que accede a la aplicación, será redirigido a la página de subscripción de cursos. En ella, encontrará un listado con los cursos en los que todavía no se ha matriculado y podrá seleccionarlos y pulsar el botón "Subscribir" para matricularse en los mismos. Una vez hecho esto, pulse el botón "Ver lecciones" para visualizar el contenido de los mismos. Si pulsa el botón "Crear Curso" podrá acceder al área de profesor para gestionar y crear nuevos cursos.

NOTA: Podrá volver a esta página en cualquier momento pulsando sobre el botón "Matricular cursos" del la Vista de Lección.

9.4. Vista de Lección

Has iniciado sesión como: **J Manuel Maldonado**

[Cerrar sesión](#)

VER LECCIONES

Asignaturas:

- ☒ TeachApp
- ☒ Historia Moderna
- ☒ Matemáticas
- ☒ Conocimiento del Medio

TEACHAPP: Teoría 1

Añadir elemento de clase **Nombre:** **Elemento:** [Añadir](#)

[Video Presentación](#)

Video Presentación

Presentación PFC TeachApp

Recursos:

- [Memoria-PFC-Jose Manuel Maldonado.docx](#)

Proximas lecciones

Fecha y hora	Asignatura	Lección
04/01/2014 10:30 - 12:30	TeachApp:	Practica 1
09/01/2014 09:00 - 11:00	TeachApp:	Teoría 2
11/01/2014 10:30 - 12:30	TeachApp:	Practica 2
07/11/2013 16:00 - 18:00	Historia Moderna:	Introducción
09/11/2013 17:00 - 20:00	Historia Moderna:	Siglo XVII

Esta página muestra los cursos suscritos (en la columna de la izquierda) y le permite seleccionarlos. Si selecciona uno o más cursos y pulsa sobre el botón “Ver cursos”, aparecerán únicamente los cursos seleccionados, de forma que podrá filtrar por asignaturas. Si desea volver a ver todos los cursos, marque o desmarque todas las opciones y vuelva a pulsar sobre “Ver cursos”.

Si pulsa sobre el botón “Matricular cursos” podrá acceder a la página descrita en el anterior apartado y podrá suscribirse a nuevas asignaturas, que aparecerán en el menú izquierdo de la ventana.

En el menú inferior aparecerán las próximas cinco clases que usted tiene pendientes de los cursos seleccionados. Podrá navegar pulsando sobre ellas, de forma que visualizará la información completa de la lección. Si hace esto, se mostrarán en este menú las lecciones inmediatamente posteriores a la que esta visualizando. Para volver al inicio de la secuencia, pulse sobre el botón “Ver Cursos” o navegue hasta que ya no haya próximas lecciones por visualizar.

En la parte superior de la ventana, podrá encontrar información acerca del curso que está visualizando así como el nombre de la lección actual. Si pasa el cursor por encima del nombre de la asignatura aparecerá información sobre la misma como la descripción o la evaluación. En el menú lateral de su derecha encontrará el día y la hora en la que se desarrollará la lección así como los recursos disponibles para la misma. Para seleccionar un recurso, pulse sobre su nombre en el menú lateral y el navegador le preguntará si desea abrirlo o guardarlo en el equipo. Pulse la opción deseada y tendrá acceso al recurso.

En el área central, el profesor podrá publicar material adicional para la clase, tal y como videos, enlaces y otros recursos que serán de utilidad durante su aprendizaje. Además, existe la posibilidad de que el profesor inicie un Hangout en directo para poder seguir en vivo sus explicaciones sobre los recursos facilitados. Una vez concluida la clase, podrá acceder a la grabación de la clase en video. También es posible enlazar a un archivo de Google Drive, donde podrá interactuar de forma colaborativa sobre el material.

Si es usted profesor del curso, además podrá visualizar un formulario que le permitirá añadir contenido de clase indicando el nombre con el que se mostrará y la URL donde este contenido (si es un video, se mostrará embebido en la propia aplicación). Pulse añadir y podrá visualizarlo en la página de la lección.

Si pulsa sobre el botón “Ver Evaluaciones” accederá a la página de calificaciones que se explica a continuación.

9.5. Evaluaciones



Has iniciado sesión como:
J Manuel Maldonado

Cerrar sesión

CALIFICACIONES

matematicas

Alumno	trabajo1	Totales
J Manuel Maldonado	15.0%	15.0%
	7.5	
	Fallaste en el punto 2	1.125 (7.5)

Ver lecciones

La página de calificaciones le mostrará los resultados de la evaluación de los distintos trabajos, ejercicios y exámenes que haya realizado. En la parte superior derecha de la ventana, podrá ver el curso sobre el que se refiere la evaluación. En la parte central, aparecerá su nombre y sucesivas columnas que indicarán el concepto evaluado así como el peso del mismo en la evaluación global. El profesor le evaluará y les asignará su nota, que podrá ir acompañada o no de un comentario. En la parte derecha podrá ver una columna de totales en el que se le indicará la nota media calculada sobre el peso de los conceptos actualmente evaluados, así como su ponderación actual sobre 10.

Para volver a la página de lecciones, pulse sobre el botón “Ver lecciones”.

Has iniciado sesión como:
J Manuel Maldonado
 Cerrar sesión

CALIFICACIONES

Facturacion y Finanzas

Nuevo concepto de evaluación
 Concepto:
 Porcentaje: %
 Añadir Concepto

Alumno	Primer Examen 50.0 %	Primer Trabajo 50.0 %	Segundo Examen 50.0 %	Segundo Trabajo 25.0 %	Tercer Trabajo 10.0 %	Prueba 10.0 %	otro 10.0 %	Totales 205.0 %
J Manuel Maldonado	0.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	0.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	0.0 Comentario privado Comentario público	9.5 (4.63)
Pepito Rodriguez	10.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	10.0 Comentario privado Comentario público	1.0 Comentario privado Comentario público	0.0 Comentario privado Comentario público	18.6 (9.07)
Mitaku Kokiri	10.0 Comentario privado Comentario público	0.0 Comentario privado Comentario público	Comentario público Excelente Trabajo!				0.0 Comentario privado Comentario público	5.0 (2.44)

Guardar Evaluaciones
 Ver Lecciones
 Ver Docencia

Si es usted profesor de la asignatura, la página será algo más compleja. Al igual que en la vista del alumno, usted podrá ver el nombre del curso en la parte superior derecha de la ventana. En la parte central, dispondrá de una lista con los alumnos suscritos a su asignatura así como sucesivas columnas con los conceptos evaluados. Para evaluar a un alumno, no tiene más que introducir su nota en la casilla correspondiente y añadir opcionalmente un comentario. Existen dos tipos de comentarios: público o privado. El primero de ellos podrá ser visualizado por el alumno mientras que el segundo solo podrá verlo usted. Para añadir un comentario, pase el cursor sobre el tipo de comentario que desee añadir en la casilla correspondiente. Aparecerá un cuadro de texto donde podrá introducir el comentario. Una vez hecho esto, debe pulsar sobre el botón “Guardar Evaluaciones” para que los datos queden registrados.

NOTA: podrá evaluar a varios alumnos y varios conceptos a la vez, pero deberá pulsar sobre el botón “Guardar Evaluaciones” o de lo contrario los cambios no quedarán registrados.

En la parte superior de la ventana el profesor dispone de un formulario para añadir un nuevo concepto. Para ello, debe añadir el nombre del concepto así como el porcentaje de peso sobre el total de la asignatura. Una vez hecho esto, pulse sobre el botón “Añadir concepto” y aparecerá una nueva columna con el concepto a evaluar. Por defecto, la nota aparece a “0”, por lo que deberá evaluar a sus alumnos sobre dicho concepto o perderán la nota sobre el mismo.

En la columna de la derecha, usted podrá visualizar la tabla de totales calculada para todos sus alumnos sobre el peso total impuesto en los conceptos y sobre 10. Recuerde que no necesariamente debe evaluar una asignatura sobre un porcentaje del 100%, sino que podrá añadir más o menos peso sobre los distintos conceptos.

Recuerde que podrá acceder a la página de calificaciones desde el área de Leccion y desde el área del profesor pulsando sobre el botón “Ver evaluaciones”.

9.6. Ver cronograma

Has iniciado sesión como: J Manuel Maldonado

VER CRONOGRAMA

Cerrar sesión

Seleccione el cronograma: Facturacion y Finanzas01032013 ▼ Mostrar

Guardar Descripciones Editar el Cronograma Evaluaciones

Descripción

Curso de facturacion y finanzas básico impartido por la Cámara de Comercio.

Evaluación

100% examen.
Habrá que asistir a clase y realizar los ejercicios y prácticas propuestas

Ir a lecciones

Podrá acceder a esta página desde la matriculación de cursos, pulsando sobre el botón “Crear Curso”. Esta es el área del profesor y en la esquina superior derecha podrá ver la asignaturas que imparte en el desplegable. Si selecciona una y pulsa el botón “Mostrar” aparecerán en el centro de la pantalla las informaciones específicas de la asignatura tales como la descripción de la misma y su evaluación. Para editar o añadir esta información escriba en el formulario el texto que desee publicar y pulse sobre el botón “Guardar Descripciones”.

Si desea editar el cronograma de una asignatura, seleccione el curso deseado y pulse sobre el botón “Editar el Cronograma”. Si por el contrario desea evaluar a sus alumnos, seleccione la asignatura y pulse sobre “Evaluaciones”.

Para crear un curso nuevo, seleccione la opción “NUEVO CRONOGRAMA” en el desplegable y a continuación pulse el botón “Mostrar” o el botón “Editar el cronograma”.

Para volver al área de lecciones pulse el botón “Ir a lecciones”.

9.7. Nuevo Cronograma

TeachApe

Has iniciado sesión como: **J Manuel Maldonado**

[Cerrar sesión](#)

CREAR UN NUEVO CRONOGRAMA

Introduzca el nombre del curso:

Introduce la fecha de inicio del curso:

¿Cuántas semanas dura el curso?

Día	Hora de Inicio	Hora de Fin	Descripción
Lunes	<input type="text"/>	<input type="text"/>	<input type="text"/>
Martes	<input type="text" value="16:00"/>	<input type="text" value="17:00"/>	<input type="text"/>
Miercoles	<input type="text"/>	<input type="text"/>	<input type="text"/>
Jueves	<input type="text"/>	<input type="text"/>	<input type="text"/>
Viernes	<input type="text" value="15:00"/>	<input type="text" value="17:00"/>	<input type="text"/>
Sabado	<input type="text"/>	<input type="text"/>	<input type="text"/>
Domingo	<input type="text"/>	<input type="text"/>	<input type="text"/>

[Generar Cronograma Preliminar](#)

En esta página usted podrá crear un nuevo cronograma para publicar un nuevo curso. Para ello, introduzca en el formulario superior el nombre del curso, la fecha de inicio del mismo y su duración en semanas.

NOTA: Estos campos son obligatorios y deberá introducirlos correctamente antes de continuar, de lo contrario aparecerá en rojo un mensaje de error junto al campo incorrecto. Recuerde que la fecha debe de tener el formato dd/MM/yyyy.

Opcionalmente, usted puede rellenar la tabla semanal situada en la parte central de la pantalla. En ella, podrá indicar el horario de las clases así como una descripción, por ejemplo: teoría o prácticas. Estos datos se usarán de patrón para generar todas las semanas del curso.

IMPORTANTE: recuerde que estos datos no son obligatorios, si lo desea podrá añadir o editar estas descripciones posteriormente. Si añade estos datos, deberá utilizar el formato de fecha hh:mm o de lo contrario le aparecerá un mensaje de error.

Una vez haya rellenado los datos del curso, pulse sobre el botón “Generar Cronograma Preliminar” tras lo cual será redirigido a la página de edición de cronogramas, donde podrá complementar la información del curso.

9.8. Editar el Cronograma

Has iniciado sesión como: **J Manuel Maldonado** [Cerrar sesión](#)

EDITAR EL CRONOGRAMA: Facturacion y Finanzas01032013

Subir un Recurso Lección: **Introduccion** Archivo: Examinar... Subir fichero

Semana 1				
Día	Hora de Inicio	Hora de Fin	Descripción	Recursos
Lunes				Ver
Martes	13:00	14:00	Introduccion	Ver
Miercoles				Ver
Jueves	18:00	20:00	Historia	Ver
Viernes				Ver
Sabado				Ver
Domingo				Ver

Recursos disponibles: [tutoria pfc.txt](#)

Semana 2				
Día	Hora de Inicio	Hora de Fin	Descripción	Recursos
Lunes				Ver
Martes	13:00	14:00	Conceptos basicos de facturacion	Ver
Miercoles				Ver
Jueves	18:00	20:00	aspectos legales de facturacion	Ver
Viernes				Ver
Sabado				Ver
Domingo				Ver

Esta página podrá ser accedida desde la página “Ver cronogramas” del profesor o tras la creación de un nuevo curso. En ella, se mostrará un detalle con todas las semanas del curso, que aparecerá en la esquina superior derecha de la pantalla.

Si pulsa sobre el botón marcado con un “.” en la parte izquierda de la ventana, usted podrá marcar el día con festivo, mostrándose en ese caso el día en rojo (como los fines de semana). Podrá utilizar esta función para marcar/desmarcar festivos en su calendario, incluidos los fines de semana.

Si usted pasa el ratón por encima del número de la semana, le aparecerá un pequeño menú que permitirá dicha semana como de “Vacaciones”. Si usted hace esto, toda la semana se teñirá de rojo en el cronograma. Si vuelve a pulsar sobre el botón “Vacaciones”, la semana volverá a mostrar los colores habituales.

Usted podrá editar la información de todos los campos mostrados, ya sea la hora de la lección (recuerde

Sabado				Ver
Domingo				Ver

Semana 10				
Día	Hora de Inicio	Hora de Fin	Descripción	Recursos
Lunes				Ver
Martes	13:00	14:00	practica con contaplus	Ver
Miercoles				Ver
Jueves	18:00	20:00	examen final	Ver
Viernes				Ver
Sabado				Ver
Domingo				Ver

[Guardar](#) [Exportar a ICS](#)

usar el formato hh/mm o le aparecerá un mensaje de error) o la descripción de la clase. Una vez editados los campos, deberá pulsar sobre el botón “Guardar” situado en la parte inferior de la página.

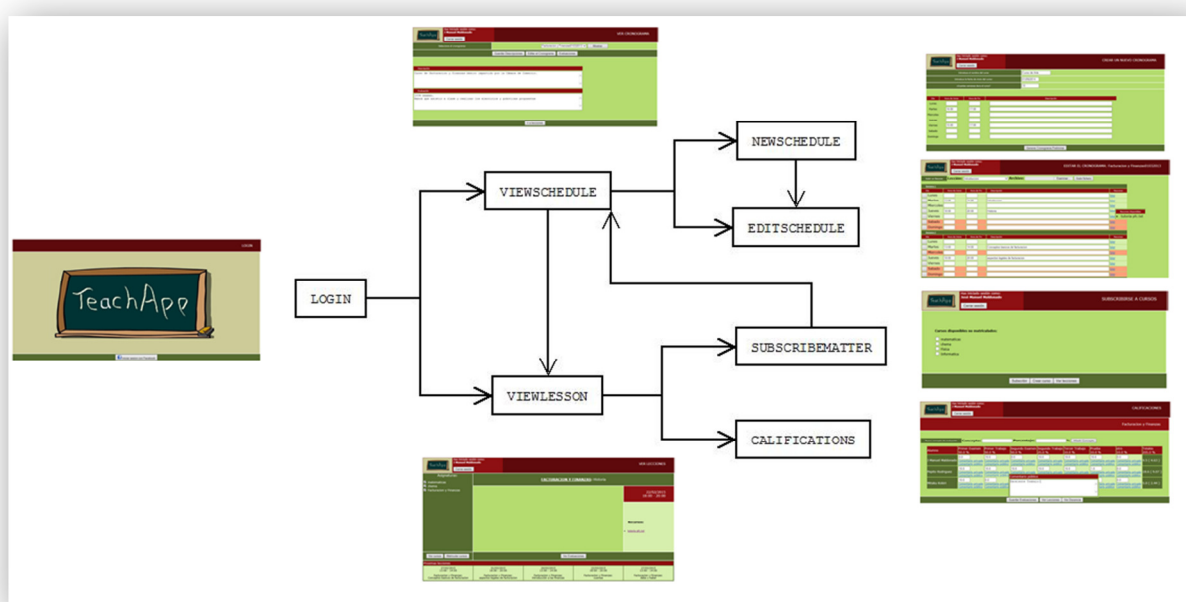
En la parte inferior, también encontrará el botón “Exportar a ICS”. Esta funcionalidad le permite añadir a su calendario habitual (Google Calendar, Outlook, etc) la información de las lecciones del curso en la fecha y horarios establecidos.

Como habrá observado, en la parte derecha del cronograma aparece una columna llamada Recursos con el texto en azul “Ver”. Si pasa el cursor sobre este texto, aparecerá una nueva área de pantalla con los recursos incluidos por usted para la lección en cuestión. Para añadir un nuevo recurso a una lección, utilice el formulario ubicado en la parte superior de la página. Para ello, seleccione la descripción del curso en el desplegable y pulse sobre el botón “examinar” para seleccionar entre sus archivos el recurso que desea compartir. Una vez hecho esto, pulse el botón “subir fichero”, el recurso aparecerá asociado a la lección correspondiente y podrá ser descargado por sus alumnos.

9.9. Resumen

TeachApp le permite crear y subscribirse a cursos con facilidad, proporcionando una plataforma adecuada para la difusión de contenidos y recursos así como su seguimiento por medio de enlaces a contenido interactivo y videos. Podrá seguir sus evaluaciones y exportar sus lecciones a su calendario habitual.

A continuación, se muestra un resumen de la navegación de la plataforma para su mayor claridad.



10. ANEXO III: Presupuesto

En este anexo se presenta el presupuesto del proyecto. En las siguientes páginas, se desglosarán los costes derivados del mismo en personal, amortizaciones y gastos directos e indirectos. Así mismo, se establecerán los tiempos y la dedicación necesaria para realizarlo y otros parámetros para el debido cálculo de costes.

10.1. ¿Qué se presupuesta?

El presente presupuesto se refiere a la aplicación TeachApp, que es una plataforma de E-Learning desarrollada con tecnología Java y orientada a la integración de tecnologías gratuitas existentes en el mercado como Facebook y DropBox. Su principal objetivo es la creación de cursos y la difusión de recursos, de forma que los alumnos puedan seguir con facilidad y a distancia el desarrollo de una materia de forma que puedan obtener un conocimiento de la misma.

Esta aplicación se enmarca en el ámbito de un proyecto de fin de carrera de la Universidad Carlos III de Madrid desarrollado por el estudiante de Ingeniería Informática José Manuel Maldonado Muñoz y tutorizado por el profesor Javier García Guzmán.

10.2. Parámetros generales

Para la realización de un proyecto de fin de carrera, se establece que un alumno deberá dedicar una media de 300 horas, por lo que partiremos de este dato para realizar los cálculos pertinentes al desglose de tiempos. Puesto que el proyecto comenzó en Diciembre de 2012 y finaliza en Septiembre de 2013, se establece una duración del mismo de 10 meses, lo cual equivale a una dedicación al mismo de 30 horas por mes. En resumen:

Horas Totales	300
Duración en Meses	10
Hora / Mes	30

Por otra parte, también debemos definir una tasa de costes indirectos, que comúnmente se establece en un 20% del coste total del proyecto.

Tasa de costes indirectos	20%
----------------------------------	-----

A partir de ahora, partiremos de estos datos para calcular los costes asociados en el presupuesto que se desglosarán a continuación.

10.3. Costes de Personal

Este documento no tiene como objeto establecer la fórmula legal de contratación del personal necesario para la realización del proyecto (autónomos o trabajadores por cuenta ajena) ni tampoco pretende desglosar los costes asociados a cada modalidad de contratación tales como seguridad social, IVA, etc. En su lugar, se partirá de los siguientes costes comúnmente aceptados en función de la categoría profesional que se requiere para el desarrollo del presente proyecto.

Categoría	Coste/Mes
Ingeniero Senior	4.289,54 €
Ingeniero Junior	2.694,39 €

Identificamos la necesidad de dos profesionales en el proyecto. Por una parte José Manuel Maldonado Muñoz tendrá una categoría de Junior, mientras que Javier García Guzmán, tendrá una categoría Senior. Se establece una dedicación de 30 horas mensuales para José Manuel, tal y como se vio en los parámetros generales. Se establece una dedicación media de 4 horas mensuales para Javier, computadas en reuniones, correcciones, asesoramiento e investigación. De esta forma:

Empleado	Categoría	Coste/Mes	Horas/Mes	Nº Meses	Total
José Manuel Maldonado	Ingeniero Junior	2.694,39 €	30	10	6.158,61 €
Javier García Guzman	Ingeniero Senior	4.289,54 €	4	10	1.307,29 €
					7.465,90 €

NOTA: *HombreMES* = 131,25 horas.

Fórmula: $(\text{CosteMES} * \text{HorasMES} / \text{HombreMES}) * \text{Nº Meses}$

10.4. Amortización de equipos

En este apartado se desglosan los equipos utilizados durante el desarrollo del proyecto. Se considerará todo aquel material utilizado específicamente para el desarrollo del mismo, por lo que se computará como coste indirecto todo aquel material que ya existiera independientemente del desarrollo del proyecto y cuyo coste ya está imputado en la realización de otras actividades habituales, tales como escritorios, sillas, alquiler de inmuebles, etc.

Descripción	Coste (sin IVA)	% Uso al proyecto	Periodo depreciación	Nº Meses	Total
Portatil Toshiba Satellite L870-10Z	578,51 €	50%	60 meses	10	48,21 €
					48,21 €

NOTA: Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado

B = periodo de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto

10.5. Subcontratación de tareas

En este apartado se incluyen todas actividades que por sus especificaciones concretas, no pueden ser desempeñadas directamente por el equipo de desarrollo y tienen que ser delegadas a terceros especializados en dichas prácticas.

Tarea	Empresa	Coste Imputable
Impresión del proyecto	Reprografía UC3M	50 €

10.6. Costes de Funcionamiento

En este apartado se engloban todos los gastos no contemplados en capítulos anteriores tales como fungibles, dietas, viajes y otros.

Se han contemplado como gastos directos los costes del transporte a las reuniones con el tutor. Puesto que el viaje se realiza en coche particular, se incluyen los costes de la gasolina asociada, no así la amortización del coche, el seguro o la provisión para revisiones y desperfectos, que serán considerados gastos indirectos de cara al proyecto. Se calcula un coste de depósito diésel a una media de 60 €, lo que permite recorrer unos 900 KM. Puesto que se requiere una reunión cada 3 semanas durante 10 meses, se estiman 14 desplazamientos. El trayecto es de 16 Km. Por tanto el porcentaje de uso para este fin es:

$(16 \text{ Km} * 14 \text{ desplazamientos} * 100) / (900 \text{ Km deposito} * 10 \text{ meses}) = 2.49\%$ del depósito al mes.

Por otra parte, se contemplan también como gastos directos los gastos de suministros consumidos por el trabajador José Manuel en su lugar de trabajo, dado que ejerce su actividad de forma específica durante unas horas al mes. Se considerarán los gastos de suministro de Javier como gastos indirectos, dado que ese gasto se imputaría independientemente del desarrollo de este proyecto en los gastos de la Universidad. Por tanto, calculamos el porcentaje de tiempo del suministro dedicado a la actividad:

$(30 \text{ HorasTrabajadas/Mes} * 100) / (30 \text{ Días/Mes} * 24 \text{ Horas/Día}) = 4.17\%$ del suministro al mes.

Descripción	Empresa	% Consumo	Meses	Precio/Mes	Total
Gasolina	Coche	2,49%	10	60 €	14,94 €
Luz	Iberdrola	4,17%	10	35 €	14,60 €
Gas	Gas Natural	4,17%	10	50 €	20,85 €
Agua	Canal Isabel II	4,17%	10	17 €	7,09 €
Internet	ONO	4,17%	10	47 €	19,60 €
					77,08 €

10.7. Resumen de costes

Tras el desglose detallado de los apartados anteriores y teniendo en cuenta que se estableció una tasa de costes indirectos del 20%, se obtiene que el coste total del proyecto es de 9169,00 € (Nueve mil ciento sesenta y nueve Euros con cero céntimos de Euro).

$7.466,00 \text{ €} + 48,00 \text{ €} + 50,00 \text{ €} + 77,00 \text{ €} = 7641,00 \text{ €}$ coste directo.

$7641,00 \text{ €} * 20 \% = 1.528,00 \text{ €}$ coste indirecto.

Presupuesto Costes Totales	Importe
Personal	7.466,00 €
Amortización	48,00 €
Subcontratación de tareas	50,00 €
Costes de funcionamiento	77,00 €
Costes Indirectos	1.528,00 €
Total	9.169,00 €